

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-050586

(43)Date of publication of application : 21.02.1995

(51)Int.Cl. H03M 7/00  
H03M 1/12

(21)Application number : 04-266900

(71)Applicant : AT & T CORP

(22)Date of filing : 10.09.1992

(72)Inventor : CHEN JUIN-HWEY

(30)Priority

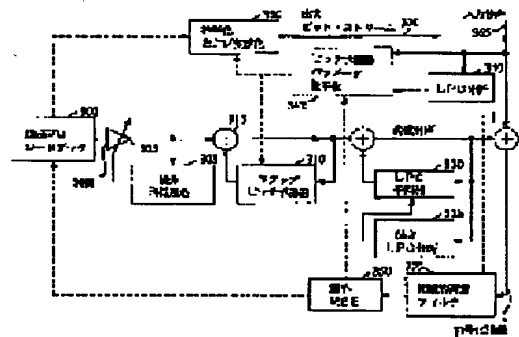
Priority number : 91 757168 Priority date : 10.09.1991 Priority country : US

## (54) LOW-DELAY CELP SPEECH ENCODING METHOD

(57)Abstract:

PURPOSE: To provide a low bit speed (generally, 8kbts/s) and low-delay digital encoder/decoder, based on code excited linear prediction for sound and similar signals.

CONSTITUTION: This method is provided with backward adaptive control for the gain of a code book and parameters of a short-term synthesis filter, as well as forward adaptive control for the parameters of a long-term (long-pitch) synthesis filter. By efficient expansion and quantization with a small delay of a pitch parameter, only a general delay corresponding to only a part of delay in a conventional encoding system is allowed to obtain an equivalent sound quality at a lower bit rate.



## LEGAL STATUS

[Date of request for examination] 06.05.1993

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2971266

[Date of registration] 27.08.1999

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-50586

(43)公開日 平成7年(1995)2月21日

(51)Int.Cl.<sup>6</sup>

H 0 3 M 7/00  
1/12

識別記号

庁内整理番号

8522-5 J

F I

技術表示箇所

審査請求 有 請求項の数36 F D (全 31 頁)

(21)出願番号 特願平4-266900

(22)出願日 平成4年(1992)9月10日

(31)優先権主張番号 7 5 7 1 6 8

(32)優先日 1991年9月10日

(33)優先権主張国 米国 (U S)

(71)出願人 390035493

エイ・ティ・アンド・ティ・コーポレー  
ション

AT&T CORP.

アメリカ合衆国 10013-2412 ニューヨ  
ーク

ニューヨーク アヴェニュー オブ  
ジ アメリカズ 32

(72)発明者 ジュアン・フエイ チェン

アメリカ合衆国 08853 ニュージャージ  
ー

サマセットカウンティ、ネシャニッ  
クステーション ロングフィールド ドラ  
イヴ 68

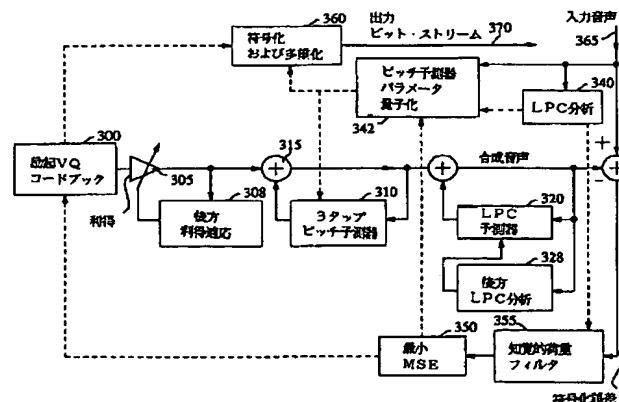
(74)代理人 弁理士 三俣 弘文

(54)【発明の名称】 低遅延CELP符号化方法

(57)【要約】

【目的】 音声および類似信号のための符号励起線形予測に基づく低ビット速度（一般に8 kbit/s）低遅延デジタル符号器／復号器を与える。

【構成】 コードブックの利得および短期合成フィルタのパラメータのための後方適応調節、ならびに長期（長ピッチ）合成フィルタのパラメータに対する前方適応調節を特徴とする。ピッチ・パラメータの高効率かつ低遅延の展開および量子化により、従来の符号化方式の遅延のごく一部に相当する総体的遅延しか許さず、低ビット・レートで同等の音声品質を得る。



## 【特許請求の範囲】

【請求項1】 Rキロビット／秒の速度で標本化される入力信号の標本からなるFミリ秒のフレームをDミリ秒の符号化遅延で符号化する方法において、それぞれのインデックス信号を有する複数のコードブック・ベクトルの各々に対し、利得調整されたベクトルを生成するために、利得因子により前記ベクトルを調整するステップと、

合成された候補信号を生成するために、前記入力信号の長期的特性を反映する長期フィルタおよび前記入力信号の短期的特性を反映する短期フィルタの従属接続に前記の利得調整されたベクトルを適用するステップと、前記フレームの標本化された入力信号を最適に近似する候補信号を決定するために、前記候補信号の各々を前記フレームの標本化された入力信号と比較する比較ステップと、

前記フレームの標本化された入力信号を最適に近似する候補信号に対応するインデックスを前記フレームの後の復号に利用できるようにするステップと、

前記の長期フィルタに対するフィルタ・パラメータを展開する長期パラメータ展開ステップと、

前記の長期フィルタに対するフィルタ・パラメータを前記フレームの後の復号に利用できるようにする長期パラメータ利用ステップと、

前記の短期フィルタに対するフィルタ・パラメータを後方適応によって展開する短期パラメータ展開ステップとを備えたことを特徴とする低遅延CELP符号化方法。

【請求項2】 前記短期フィルタが、20に満たないNS個のフィルタ・タップを有するフィルタであり、かつ前記の短期パラメータ展開ステップが、前記のNS個のタップの各々に対する係数の値を展開するステップを含むことを特徴とする請求項1記載の方法。

【請求項3】 Fが5以下であることを特徴とする請求項1記載の方法。

【請求項4】 Dが10以下であることを特徴とする請求項1記載の方法。

【請求項5】 Rが16未満であることを特徴とする請求項4記載の方法。

【請求項6】 前記利得因子が、後方適応によって調整されることを特徴とする請求項2記載の方法。

【請求項7】 前記比較ステップが、各候補信号に対し、前記入力フレームと前記候補信号との間の差を表す差信号を形成するステップと、知覚的により重要な周波数を強調するように加重された差信号を形成するために、前記の差信号を周波数荷重するステップと、

前記の加重された差信号に対する最小値を決定するステップとを備えたことを特徴とする請求項1記載の方法。

【請求項8】 その係数が入力フレーム信号の分析によ

って決定されるようなフィルタに前記の差信号を通すことによって、前記の周波数荷重を実現することを特徴とする請求項7記載の方法。

【請求項9】 前記の入力フレーム信号の分析が、量子化されていない入力フレーム信号のLPC分析からなることを特徴とする請求項8記載の方法。

【請求項10】 NS=10であることを特徴とする請求項2記載の方法。

【請求項11】 前記の長期パラメータ展開ステップが、ピッチ周期（または間隔）パラメータおよびNL>1なるNL個のフィルタ・タップの係数パラメータを展開するステップを含むことを特徴とする請求項1記載の方法。

【請求項12】 NL=3であることを特徴とする請求項11記載の方法。

【請求項13】 標本化された入力信号の前記フレームが、発声された情報列の一部であるか否かを判断する判断ステップと、

標本化された入力信号の前記フレームが、発声された情報列の一部であるとき復号のために、前記長期フィルタに対する前記フィルタ・パラメータを利用できるようにするステップをさらに備えたことを特徴とする請求項1記載の方法。

【請求項14】 前記判断ステップが、各フレームに対して発声／非発声の予備的な判断を行う予備判断ステップと、

現在のフレームおよび所定の数Kの直前のフレームの各々に対する前記の予備的な判断が非発声である場合、現在のフレームが発声された音声フレームの列の一部ではないと判断するステップとを備えたことを特徴とする請求項13記載の方法。

【請求項15】 前記の予備判断ステップが、入力フレームにある標本にしきい値を確立するステップと、

現在の標本に対する値が既存のしきい値に等しいか、それ以下である場合は常に、 $T < 1$ である所定の因数Tを既存のしきい値に乘じるサブステップ、および現在の標本の値が既存のしきい値を超える場合、しきい値を現在の標本の値に設定するサブステップによって、入力フレームにおいて連続する各標本に対して前記しきい値を調節するステップと、

各入力フレームに対し、そのフレームにおける標本に対するしきい値に基づいて基準値を形成する基準値形成ステップと、

現在のフレームの標本に対する値が前記基準値に関係する第1の所定の条件を満たす場合は常に、現在のフレームが発声されたものであるとの判断を下すステップと、現在のフレームの標本に対する値が前記基準値に関係する第2の所定の条件を満たす場合は常に、現在のフレー

ムは非発声であるとの予備的な判断を下すステップとを備えたことを特徴とする請求項14記載の方法。

【請求項16】 前記基準値形成ステップが、現在のフレームの標本に対し、しきい関数の平均を形成するステップを含み、  
前記の第1の所定の条件が、前記基準値の半分を超える現在のフレームの標本に対する最大強度を有することからなり、  
前記の第2の所定の条件が、前記基準値の2%を超えない現在のフレームの標本に対し最大強度を有することからなり、さらに前記方法が、  
タップが1つの予測器に対する最適タップ値を現在の入力フレームに基づいて決定するステップと、  
前記の第1および第2の所定の条件が満たされず、前記のタップが1つの予測器のタップ値が所定の値より大きい場合、前記の現在のフレームは発声されたものであると判断するステップとを備えたことを特徴とする請求項15記載の方法。

【請求項17】 前記基準値形成ステップが、現在のフレームの標本に対し、しきい関数の平均を形成するステップを含み、  
前記の第1の所定の条件が、前記基準値の半分を超える現在のフレームの標本に対する最大強度を有することからなり、  
前記の第2の所定の条件が、前記基準値の2%を超えない現在のフレームの標本に対し最大強度を有することからなり、さらに前記方法が、  
現在のフレームの標本の正規化された1次自己相関係数を決定するステップと、  
前記の第1および第2の条件が満たされず、前記の自己相関係数が所定の値より大きい場合、前記の現在のフレームは発声されたものであると判断するステップとを備えたことを特徴とする請求項15記載の方法。

【請求項18】 前記基準値形成ステップが、現在のフレームの標本に対し、しきい関数の平均を形成するステップを含み、  
前記の第1の所定の条件が、前記基準値の半分を超える現在のフレームの標本に対する最大強度を有することからなり、  
前記の第2の所定の条件が、前記基準値の2%を超えない現在のフレームの標本に対し最大強度を有することからなり、さらに前記方法が、  
現在のフレームの標本全体に対するゼロ交差率を決定するステップと、  
前記の第1および第2の条件が満たされず、前記のゼロ交差率が所定の値より大きい場合、前記の現在のフレームは発声されたものであると判断するステップとを備えたことを特徴とする請求項15記載の方法。

【請求項19】  $K=3$ であることを特徴とする請求項14記載の方法。

【請求項20】 前記長期フィルタに対するピッチ周期を展開するステップが、  
入力フレームの信号のL次LPC分析を行うステップと、

予測残差信号を決定するために、前記L次LPC分析により展開したフィルタ係数に基づいて前記入力フレーム信号の逆LPCフィルタ処理を行うステップと、  
前記の予測残差信号の関数の相関ピーク採取によって前記ピッチ周期を抽出する抽出ステップとを備えたことを特徴とする請求項11記載の方法。

【請求項21】 前記予測残差信号の前記関数が、前記予測残差信号の時間削減され低域通過フィルタを通された関数であることを特徴とする請求項20記載の方法。

【請求項22】 前記の相関ピーク採取が、可能なピッチ周期の持続時間の範囲にわたる時間遅れに対して行われ、かつ前記抽出ステップが、最大の相関を与える時間遅れを選択するステップを含むことを特徴とする請求項20記載の方法。

【請求項23】 前記の相関ピーク採取が、可能なピッチ周期の持続時間の範囲にわたる時間遅れに対して行われ、かつ前記抽出ステップが、  
最大の相関を与える時間遅れを選択するステップと、  
ピッチ周期の値 $p_0$ を与えるために、前記の選択された時間遅れを前記時間処理を補償するように調節するステップとを備えたことを特徴とする請求項21記載の方法。

【請求項24】 基準値として前の周期に対して決定されたピッチ周期を確立するサブステップ、および前記のピーク採取において前記基準値の予め選択された範囲にあるピークによって示されるピッチ周期値 $p_1$ を現在のフレームに対して選択するサブステップによって、真のピッチ周期の不正な倍数を前記の調節された時間遅れから除去するステップをさらに備え、かつフレーム列において重要なピッチ成分を有する最初のフレームに対する基準値が、前のピッチ周期値を参照することなく前記相関関数のピークとして選択されることを特徴とする請求項23記載の方法。

【請求項25】 ピッチ周期 $p_0$ を有する入力フレームに基づいて単一タップ予測器に対する最適タップ荷重を決定し、さらにそれを0と1との間の範囲に正規化することによって、値 $W_0N$ を形成するサブステップと、  
ピッチ周期 $p_1$ を有する入力フレームに基づいて単一タップ予測器に対する最適タップ荷重を決定し、さらにそれを0と1との間の範囲に正規化することによって、値 $W_1N$ を形成するサブステップと、  
 $W_1N$ が $W_0N$ の所定の割合に等しいか、またはそれ以上である場合、正しいピッチ推定値として $p_1$ を選択し、そうでない場合、正しいピッチ推定値として $p_0$ を選択するサブステップとによって、前記所定の範囲にあるピッチ周期 $p_1$ の値と前記所定の範囲の外にあるピッチ

周期  $p_0$  の値との間で起こり得る競合を解消するステップをさらに備えたことを特徴とする請求項 2 4 記載の方法。

【請求項 2 6】 前記の所定の割合が、ほぼ 0.4 に等しいことを特徴とする請求項 2 5 記載の方法。

【請求項 2 7】 前記の長期パラメータ利用可能ステップが、

現在のフレームの入力標本からピッチ周期の第 1 の推定値を生成するステップと、

前記のピッチ周期の第 1 の推定値を丸めた表現  $r$  を生成するステップと、

入力フレームの信号の  $L$  次  $LPC$  分析を実行し、予測残差信号を決定するために、 $L$  次  $LPC$  分析によって展開したフィルタ係数に基づいて前記入力フレーム信号の逆  $LPC$  フィルタ処理を実行し、さらに前記予測残差信号の関数の相関ピーク採取によってピッチ周期の第 2 の推定値を抽出する開ループ・ステップによって、前記第 2 のピッチ周期推定値を生成するステップと、

前記第 2 のピッチ周期推定値と前記のピッチ周期の第 1 の推定値を丸めた表現との間の差を表す差信号を形成するステップとを備え、

前記差信号が、予め選択された値より大きい強度を有するときには、さらに前記差信号を複数  $q$  の所定の値の中の 1 つに量子化するステップと、

前記ピッチ周期に対して量子化された値  $p$  を、 $p = r + q$  にしたがって形成するステップとを備え、また、

前記差信号が、前記の予め選択された値に等しいか、またはそれに満たない強度を有するときには、さらに前記のピッチ周期の値の量子化を開ループ量子化方法で最適化するステップを備えることを特徴とする請求項 1 1 記載の方法。

【請求項 2 8】 前記のピッチ周期の第 1 の推定値を生成するステップが、

前記入力フレームに基づいて開ループ・ピッチ予測を形成するステップを含むことを特徴とする請求項 2 7 記載の方法。

【請求項 2 9】 前記の開ループ・ピッチ予測を形成するステップが、

前記入力フレームが発声された情報を表す標本から構成されているかどうかを判断するステップと、

前記入力フレームが発声された入力情報から構成されていない場合、前記のピッチ周期の第 1 の推定値を所定の値に設定する推定値設定ステップとを備えたことを特徴とする請求項 2 8 記載の方法。

【請求項 3 0】 前記の推定値設定ステップが、前記のピッチ周期の第 1 の推定値をピッチ周期の予測される範囲の下限から約 10% と 50% との間の値に設定するステップを含むことを特徴とする請求項 2 9 記載の方法。

【請求項 3 1】 前記のピッチ周期パラメータを展開するステップが、

前記入力フレームに基づく予測を用いて前記のピッチ周期の第 1 の推定値を形成するステップと、

直前のフレームに対するピッチ周期の予測に基づいて第 2 の推定値を形成するステップと、

前記の第 1 および第 2 の推定値の間の差を表す差信号を形成するステップと、

前記差信号が、所定の値より大きい場合、量子化された差信号を形成するために、前記の差の値を固定された複数の値の中の 1 つへと量子化するステップと、

前記第 2 の推定値および前記の量子化された差信号の和から前記ピッチ周期を得るステップとを備えたことを特徴とする請求項 1 1 記載の方法。

【請求項 3 2】 前記の第 2 の推定値を形成するステップが、

直前のフレームに対する予測値の値を遅らせるステップと、

バイアスで調節した値を与えるために、前記の遅らせた値から固定されたピッチ・バイアス値を減ずるステップと、

大きさを調節した値を形成するために、前記のバイアスで調節した値の大きさを調節するステップと、

予測ピッチ周期信号を形成するために、前記の固定されたピッチ・バイアス値を前記の大きさを調節した値に加えるステップとを備えたことを特徴とする請求項 3 1 記載の方法。

【請求項 3 3】 丸められた予測ピッチ値を形成するために、前記の予測されたピッチ周期信号を丸めるステップをさらに備えたことを特徴とする請求項 3 2 記載の方法。

【請求項 3 4】 前記の長期パラメータ展開ステップが、

前記の入力信号のフレームが、発声された情報を表さない場合、入力信号の特定の値に依存しない固定された所定の値に前記フィルタ・パラメータを設定するステップを含むことを特徴とする請求項 1 3 記載の方法。

【請求項 3 5】 前記の長期パラメータ展開ステップが、

前記のピッチ周期パラメータを、発声された情報を含む入力フレームに対するピッチ周期の値の予測される範囲の下限から約 10% と 50% との間の値に設定するステップを含むことを特徴とする請求項 3 4 記載の方法。

【請求項 3 6】 前記の入力信号のフレームが、発声された信号を表さない場合、フィルタ・タップ係数をゼロの値に等しく設定するステップをさらに備えたことを特徴とする請求項 3 5 記載の方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、伝送および記憶のための音声と関連信号の効率的符号化、および元の信号を効率的かつ忠実に再生するために後に行われる復号の分野

に関する。

#### 【0002】

【従来の技術】遠隔地との間で音声の通信を行ったり、後で取り出して再生できるように音声情報を記憶したりするために与えなければならない情報の量を減らすべく多くの方法が近年開発されてきた。考慮すべき重要な点は、そのような符号情報が符号化方式の高い品質要求に適切に応えるように生成されるべきビット・レートである。例えば、重要な用途には、毎秒32キロビット(kbit/s、以降「kbps」と記す)の割合で発生するデジタル信号によって音声が表示されるものもある。勿論、記憶または伝送の帯域幅の必要条件を最小にするためには、可能な限り少ないデジタル信号で音声を表現することが望ましい。

【0003】現在使用されている最も一般的な方式は、一括して線形予測符号化方式として周知のものである。この広い範疇の符号化方式の中で、符号励起線形予測(CELP=code excited linear prediction)符号化として周知のものが、近年多くの注目を集めている。CELP方式の初期の概要が、「音響、音声、信号処理に関するIEEE国際会議(IEEE Int. Conf. Acoust., Speech, Signal Processing)」会報p.937-p.940(1985年)のM. R. シュレーダー(Schroeder)およびB. S. アタル(Atal)による「符号励起線形予測——非常に低いビットレートで高品質音声(Code Excited Linear Prediction (CELP): High-Quality Speech at Very Low Bit Rates)」にある。

【0004】前記の他に多くの環境において起こる符号化上の制約は、音声符号化の実行に要する遅延である。従って、例えば、遅れの小さい符号化を行うことは、エコーの影響を少なくして通信リンクにおけるエコー・サブレッサへの要求を小さくするためには非常に有効である。さらに、セルラ通信システムのような環境下では、許される全体の遅れが限られていて、チャネルの符号化の遅れがチャネルのエラー制御の要点であるため、利用可能な全体の遅延「資源」を最初の音声符号化に消費しないことが非常に望ましい。

【0005】現在まで、16 kbpsまたはそれ以下で使用するほとんどの音声符号器において、なんとか良好な音声品質を達成しようと大きなブロックの音声標本をバッファで緩衝している。この標本ブロックには、一般に、約20ミリ秒の期間にわたる音声標本が含まれ、バッファで緩衝される音声の冗長性を利用するために周知の変換、予測、またはサブバンドの技術が応用できるようになっている。しかし、バッファの緩衝遅延に処理による遅延およびビット伝送遅延が加わり、通常の符号器の一方方向の全体的な符号化遅延は、一般に約50乃至60msである。勿論、このような長い遅延は、多くの用途において望ましくなく、許されるべくもない。

【0006】国際標準のグループは、最近の目標を16

kbpsの音声符号化のための低遅延CELP符号化の問題に焦点を合わせている。CCITT(国際電信電話諮問委員会)研究グループXVIIIの1988年6月の16 kbps音声符号化に関する特別グループの参照条項(付録1から案件U/XV)(CCITT Study Group XVIII, Terms of reference of the ad hoc group on 16 kbit/s speech coding(Annex I to Question U/XV))を参照せよ。CCITTのグループによって課せられた条件では、符号化遅延は2msを目標とし5msを超えてはならないということであった。このCCITTにより課せられた課題に対する解決策は、例えば次の文献にある。

「IEEE地球圏通信会議会報(Proc. IEEE Global Commun. Conf.)」p.1237-p.1241(1989年11月)のJ. H. チェン(Chen)による「16 kbpsの頑丈な低遅延CELP音声符号器(A robust low-delay CELP speech coder at 16 kbit/s)」、「音響、音声、信号処理に関するIEEE国際会議会報」p.453-p.456(1990年4月)のJ. H. チェン(Chen)による「一方方向の遅延が2ms以下の高品質16 kbps音声符号化(High-quality 16 kbit/s speech coding with a one-way delay less than 2ms)」、および「音響、音声、信号処理に関するIEEE国際会議会報」p.181-p.184(1990年4月)のJ. H. チェン(Chen)、M. J. メルクナー(Melchner) R. V. コックス(Cox)およびD. O. ボウカー(Bowker)による「16 kbps低遅延CELP音声符号器の実時間動作(Real-time implementation of a 16 kbit/s low-delay CELP speech coder)」。

【0007】最近、CCITTは、さらに一步進めて8 kbpsの音声符号化アルゴリズムの標準化を計画した。やはり、候補となるアルゴリズムは、すべて遅延時間が短いことが要求されるが、この場合、一方方向の遅延の必要条件是、約10msへと幾分緩和されている。

【0008】8 kbpsの場合、低遅延で高品質を達成することは16 kbpsの場合より難しい。これは、一部には、現在の低遅延CELP符号器がその予測器の係数を前に符号化された音声に基づいて更新する、いわゆる「後方適応」方式だからである。例えば、在ニュー・ジャージーのプレントイス・ホール社(Prentice-Hall)(1984年)発行のN. S. ジャイアント(Jayant)およびP. ノル(Noll)による「波形のデジタル符号化(Digital Coding of Waveforms)」が参考になる。さらに、8 kbpsで符号化された音声の方が符号化雑音レベルが高いので、後方適応が16 kbpsの場合より非効率的になる。

【0009】CCITTによって発表された8 kbps低遅延符号器の課題の前には、その主題に関する文献は殆どあるいは全くなかった。その発表以後、T. モリヤ(Moriya)が、発話言語の処理に関する国際会議の議事録(1990年11月)の「コンディショナル・ピッチ予測に基づく中程度の遅延の8 kbps音声符号器(Mediu

m-delay 8 kbit/s speech coder based on conditional pitch prediction)」において、例えば前記の1989年のチェンの論文において記述されている16 kbpsの低遅延CELP符号器の後方適応方式に基づく遅延時間10msの8 kbpsCELP符号器を提案した。報告によれば、この8 kbps符号器の性能は、前記のシュレーダーおよびアタルによる1985年の論文、および「音響、音声、信号処理に関するIEEE国際会議会報p.1650-p.1654 (1987年)のP. クルーン (Kroon) およびB. S. アタル (Atal) による「4. 8 kbpsCELP符号器のための量子化手順 (Quantization procedure for 4.8 kbps CELP coders)」に記述されている通常の8 kbpsCELP符号器を上回ると言われている。しかし、その性能が可能なのは、(計算が極めて複雑になることを代償として) 励起ベクトルの遅延決定符号化が使用される場合に限られる。これに対して、遅延決定が使用されない場合、音声品質が、低下して通常の8 kbpsCELPより幾分劣るようになる。

【0010】モリヤの符号器では、最初に8ピッチの候補を決定するために後方適応ピッチ分析を行ったうえで、3ビットを送って選択された候補を指定した。後方ピッチ分析がチャンネル・エラーに対して非常に敏感であることは周知であるから(前記のチェンによる1989年の文献が参考になる)、この符号器もチャンネル・エラーに対して敏感のようである。

【0011】

【発明が解決しようとする課題】本発明により、従来の技術とは異なる方法を用いることにより、従来の符号器の潜在的な制限および過敏性の多くを避けながら低ビット・レート低遅延の符号化および復号を与えることである。本発明によって処理された音声は、従来のCELPの場合と同質のものであるが、従来のCELPの僅か1/5程度の遅延でそのような音声を与えることができる。さらに、全二重の符号器が単一のデジタル信号処理(DSP)チップ上に好ましい形で実施できるように、本発明では、従来の技術の複雑さの多くを回避している。さらには、本発明の符号化および復号の方式を用いることにより、ビット誤り率が高い条件の下でも双方向の音声通信を容易に達成することができる。

【0012】

【課題を解決するための手段】前記の結果は、CELP符号器における本発明の説明用の実施例において得られる。実施例では、励起利得因子および短期(LPC)予測器をいわゆる後方適応を用いて更新する。この点において、この実施例は、先に引用した論文に記述された16 kbps低遅延符号器との類似点がある(ほか、それとの重要な相違点もある)。しかし、この実施例では、より高い音声品質およびチャンネル・エラーに対するより優れた頑丈さを実現するために、重要なピッチ・パラメータは、すべて前方に送られる。

【0013】本発明の典型的な実施例に有利に使用されているピッチ予測器は、3タップ・ピッチ予測器であり、内部フレーム予測符号化方式を用いてピッチ周期(またはピッチ間隔)を符号化し、閉ループ・コードブック探索によって3つのタップをベクトル量子化する。「閉ループ」は、本明細書で用いる場合、符号化された音声の知覚的に荷重した平均2乗誤差をコードブック探索により最小にしようすることを意味する。この方法は、ビットを節約し、高いピッチ予測利得(一般に5乃至6dB)を与え、かつチャンネル・エラーに対して頑丈であることが解った。ピッチ周期は、閉ループ探索方法および閉ループ探索方法の組み合わせによって都合良く決定される。

【0014】前述の16 kbps低遅延符号器で使用された後方利得適応は、本発明の説明のための実施例においても有利に使用されている。また、従来のCELPの実施において使用された15乃至30msに比較して小さい間隔(例えば、僅か2.5乃至4.0ms)を表すフレームを用いることが有利であることも解った。

【0015】典型的な実施例に関する以下の詳細な説明において説明するその他の改良点には、閉ループ・トレーニング法によって獲得されたベクトルによる励起コードブックの導入も含まれる。

【0016】音声品質をさらに高めるために、本発明を説明する実施例の復号器には後置フィルタ(例えば、カリフォルニア大学サンタ・バーバラ校のJ. H. チェンによる博士論文「音声波形のベクトル量子化に基づく低ビット・レート予測符号化 (Low-bit-rate predictive coding of speech waveforms based on vector quantization)」に提案されたものと類似のもの)を有利に使用している。さらに、短期後置フィルタおよび長期後置フィルタを共に使用する方が有利であることが分かる。

【0017】

【実施例】本発明のより良く容易に理解できるように、通常のCELP符号器の概要を簡単に説明する。次に、(要素およびシステムのレベルで)本発明によって与えられる新機軸を説明し、最後に、本発明の一般的な説明用の実施例を詳細に説明する。

【0018】通常のCELP(符号励起線形予測)の概要

概観すると、図1のCELP符号器は、励起シーケンスを励起コードブック100から利得調整要素105を経て長期合成フィルタおよび短期合成フィルタの縦続接続へと渡すことにより、音声を合成する。長期合成フィルタは、長期予測器110および総和器要素115からなり、短期合成フィルタは、短期予測器120および総和器125からなる。当分野において周知のとおり、両方の合成フィルタは、一般に全極フィルタであり、それぞれの予測器が指示された閉ループ内に接続されている。

【0019】長期および短期の合成フィルタの縦続接続

の出力が、前記の合成された音声である。この合成された音声は、比較器130において、一般にデジタル化標本のフレームの形式で入力音声と比較される。合成および比較の動作は、コードブック100における励起シーケンスの各々に対して繰り返され、最も一致するシーケンスのインデックスが、システム・パラメータについての付加的な情報と共に後の復号に使用される。基本的に、CELP符号器は、各フレームに対し、入力音声と合成音声との間の知覚的に加重した平均2乗誤差(MSE)が最小になるような最良の予測器、利得、および励起を発見するように努めて、音声をフレーム毎に符号化する。

【0020】長期予測器は、その主な機能が発声された音声におけるピッチの周期性を利用することであることから、しばしばピッチ予測器と呼ばれる。一般に、1タップのピッチ予測器が用いられるが、この場合、その予測器の伝達関数は、 $P_1(z) = \beta z^{-p}$ である。ただし、 $p$ は群遅延、即ちピッチ周期であり、 $\beta$ は予測器のタップである。短期予測器は、2.4 kbpsまたはそれ以下のビットレートで動作する周知のLPC(線形予測符号化)ボコーダでも使用されるので、LPC予測器と呼ばれることもある。このLPC予測器は、一般に、伝達関数

【数1】

$$P_2(z) = \sum_{i=1}^{10} a_i z^{-i}$$

を有する10次予測器である。励起ベクトル量子化(VQ)コードブックには、等しい長さのコードブック・ベクトル(即ちコードベクトル)のテーブルが収容されている。一般に、コードベクトルは、可能な中央クリッピングを有するガウス乱数が占める。

【0021】さらに具体的には、図1のCELP符号器において、在ニュー・ジャージー州エンゲルウッド・クリフスのブレンティス・ホール社のL. R. ラビナ(Rabiner)およびR. W. シェイファ(Schafer)による「音声信号のデジタル処理(Digital Processing of Speech Signals)」(1978年)に概説されている種類の線形予測分析(LPC分析)を入力信号に対して最初に行うことによって、音声波形の標本をフレーム(固定長の各フレームは一般に15乃至30msの長さである)毎に符号化する。次に、結果的に得られたLPCパラメータを標準の開ループの要領で量子化する。図1では、LPC分析および量子化を要素140によって表した。

【0022】図1による標準的なCELP符号化においては、各音声フレームをそのフレーム内部で4乃至8msの間隔で発生する標本を含むいくつかの等しい長さのサブフレームまたはベクトルに分割する方が有利であることが解った。量子化されたLPCパラメータは、通

常、各サブフレームに対して補間されて、LPC予測器の係数へと変換される。そして、各サブフレームに対して、1タップのピッチ予測器のパラメータが開ループ量子化される。一般に、ピッチ周期は、7ビットに量子化され、ピッチ予測器のタップは、3または4ビットに量子化される。次に、励起VQコードブックの中の最良のコードベクトル、および最良の利得を、やはり開ループ量子化によって各サブフレームに対してフィルタ155で知覚的に荷重された入力に基づいて、最小平均2乗誤差(MSE)要素150によって決定する。

【0023】各サブフレームの量子化LPCパラメータ、ピッチ予測器のパラメータ、利得および励起コードベクトルは、図1の符号器/マルチプレクサ160によって、ビットに符号化されて、出力ビット・ストリームへと多重化される。

【0024】図2に示したCELP復号器は、音声をフレーム毎に復号する。図2において要素200で示したように、この符号器では、まず入力ビット・ストリームを分離(デマルチプレクス)して、LPCパラメータ、ピッチ予測器パラメータ、利得、および励起コードベクトルを復号する。次に、各サブフレームに対してデマルチプレクサ200によって識別された励起コードベクトルを、利得要素215における対応する利得因子によって倍率調整した後、縦続接続された長期合成フィルタ(長期予測器220および総和器225からなる)および短期合成フィルタ(短期予測器230および総和器235からなる)を通して、復号された音声を得る。

【0025】適応後置フィルタは、例えば「音響、音声、信号処理に関するIEEE国際会議」会報ASSP-29(5)のp.1062-p.1066(1987年10月)のJ. H. チェン(Chen)およびA. ガーショウ(Gercho)による「適応後置フィルタを用いた48000bpsでの実時間ベクトルAPC音声符号化(Real-time vector APC speech coding at 48000 bps with adaptive postfiltering)」において提案された種類のものであり、知覚上の音声品質を高めるために出力において一般に使用される。

【0026】前記のように、LPCパラメータは、CELP符号器によって、入力音声から直に決定され、開ループ量子化されるが、ピッチ予測器、利得、および励起は、すべて閉ループ量子化によって決定する。これらのすべてのパラメータは、符号化されてCELP復号器に送られる。

【0027】低ビット・レート低遅延の符号励起線形予測(CELP)の概略

図3および4に、本発明による低遅延符号励起線形予測(LD-CELP)符号器および復号器の実施例の概略をそれぞれ示す。便宜上、この実施例は、8 kbps LD-CELPのシステムと方法に関するCCITTの研究における切実な課題の点から説明する。しかし、ここで



説明するアルゴリズムと技術は、異なる個々のビット・レートおよび符号化遅延で動作するシステムおよび方法に同じように適用される。

【0028】図3において、入力365に現れる都合よくフレームに区切られた標本の型の中の入力音声は、ベクトルを励起コードブック300から利得調整器305と直列の長期合成フィルタと短期合成フィルタとを通過させることによって発生する合成音声と、比較器341において再び比較される。図3の実施例において、利得調整器は、以下においてさらに十分に説明するように後方適応利得調整器とみなされる。長期合成フィルタは、実例のように総和器315を有する帰還ループにおける3タップのピッチ予測器からなる。ピッチ予測器の機能は、さらに詳細に後述する。短期合成フィルタは、総和器325を有する帰還ループにおける10タップの適応後置LPC予測器320を含む。要素328で現される後方適応機能については、以下においてさらに説明する。

【0029】コードブック・ベクトルに対する平均2乗誤差の計算は、フィルタ355経由で供給された知覚的に荷重された誤差信号に基づいて、要素350において決定される。ピッチ予測器310において値を設定するために使用されるピッチ予測器パラメータ量子化は、要素342において実現される。これについては、さらに詳細に後述する。図3に示した低遅延CELP符号器の実施例の各要素間の相互関連の様子は、以下においていくつかの要素をさらに十分説明するとともに明らかになる。

【0030】図4に示した低遅延CELP符号器の実施例は、図3の実施例の符号器に対して、補完的な形で作用する。さらに具体的には、入力405で受信される入力ビット・ストリームは、要素400において復号・分離され、必要なコードブック要素の識別情報を励起コードブック410に与えるほか、ピッチ予測器のタップおよびピッチ周期情報も3タップのピッチ予測器420と総和器425からなる長期合成フィルタに与える。また適応後置フィルタ・アダプター440のための後置フィルタ係数情報も要素400によって供給される。本発明によれば、後置フィルタ445は、長期および短期の後置フィルタ処理機能を備えている。このことは、以下にさらに十分に説明する。出力音声は、後置フィルタ445を通過後、出力450に現れる。

【0031】また、図4における復号器は、総和器435を備えた帰還ループにおいて接続されたLPC予測器430（一般には、10タップ予測器）からなる短期合成フィルタも含む。短期フィルタ係数の適応化は、要素438による後方適応LPC分析を用いて、行われる。

【0032】図1および2に関連した従来のCELP符号器に関する以上の説明から、一般に従来のCELP符号器は、長期および短期のフィルタの情報、励起利得情

報、励起ベクトル情報を、これらの符号化成分のすべてに対する前方適応を可能にするために、復号器に送る。前述のチェンの論文で記述されているCCITT16 kbps低遅延CELPの必要条件の解法は、励起を除いたすべての符号情報に後方適応を用いることによって通常解決されることを示している。これらの16 kbps低遅延符号器において、ピッチ情報は明示的には使用されない。

【0033】しかしながら、図3、4から分かるように、本発明による低遅延低ビットレートの符号器／復号器は、一般にピッチ予測器パラメータと励起コードベクトル・インデックスを前方に送る。この復号器は、後方適応を利用して、直前に量子化された信号から利得およびLPC予測器を局部的に得ることができるので、それらは送る必要がないことが分かった。

【0034】従来のCELP、16 kbps低遅延CELPおよび本発明による低遅延CELPの違いを簡単に要約したので、以下の節では、本発明の実施例の個々の要素をより詳細に説明する。

#### 【0035】LPC予測

一般的な応用では、10msまたはそれ以下の一方向符号化遅延を達成するために、CELP符号器は、3乃至4ms、即ち8kHzの標本化速度では24乃至32の音声標本より大きいフレーム・バッファ・サイズをとることはできない。符号化遅延と音声品質の間のトレードオフを吟味して、2つの8 kbpsLD-CELPアルゴリズムを考えたことが好都合である。第1の例では、32標本（4ms）のフレーム・サイズで約10msの一方向遅延であったが、第2の例では、20標本（2.5ms）のフレーム・サイズで約7msの遅延であった。

【0036】8 kbpsまたは、1標本あたり1ビットでは、各フレームで使われるのは、20または、32ビットだけある。CELP符号化において、良い音声品質を得るためには励起符号化でビットの大部分を使うことが重要であるため、LPCパラメータおよびピッチ・パラメータのような非励起情報のためには、ほんの僅かしかビットが残されていないことになる。

【0037】従って、低遅延という制約（故にフレーム・サイズの制約）があるので、前述のチェンによる1989年の論文で記述されているように、後方適応によってLPC予測器の係数を更新することが好都合となる。このようなLPCパラメータの後方適応は、LPCパラメータを細かく指定するためのビット送信を必要としない。このことは、前に引用されたモリヤの論文において記述された方法と比較されるべきである。このモリヤの論文において、有望とは言えないものの、一部後方、一部前方適応構造が、LPCパラメータ適応化のために提案された。

【0038】16 kbps低遅延LD-CELPで利用される後方適応LPCパラメータの方法は、都合良く存続

されているので、8 kbpsで利用するために、16 kbps LD-CELPアルゴリズムにおいて利用されるパラメータを単に変更しようとしても当然である。この規模を小さくする方法による実験では、理解はできるが、意図した目的のためには雑音が多すぎるという結果となった。このように、本発明の実施例は、ピッチ情報の明確な誘導とピッチ予測器の使用を特徴とする。符号化および復号の動作においてピッチ予測器を利用することの重要な利点は、16 kbps低遅延法において利用される短期予測器が、一般的に従来の50タップのLPC予測器からさらに単純な10タップLPC予測器へと簡易化できるということである。

【0039】図3、4における構造で利用される説明のための10タップLPC予測器は、前述のラビナとシェイファの文献において記述されたLPC分析の自己相関法を利用してフレームごとに更新される。標準的なAT&T DSP32Cデジタル信号処理チップを使用する便利な浮動小数点の実施において、自己相関係数は、「音響、音声、信号処理に関するIEEE国際会議会報」p.453-p.456（1990年4月）のJ. H. チェンによる「2ms以下の高品質16 kbps音声符号化」および「音響、音声、信号処理に関するIEEE国際会議事録, ASSP-29(5)」p.1062-p.1066（1981年10月）のT. P. バーンウェル, IIIの「LPC分析の自己相関係数を生成するための循環窓化」に記述された修正バーンウェル循環窓を用いて、計算される。固定小数点を実施する場合、「音響、音声、信号処理に関するIEEE国際会議会報」p.21-p.24（1991年5月）のJ. H. チェン、Y. C. リン(Lin)とR. V. コックス(Cox)の「固定小数点16 kbps LD-CELPアルゴリズム」に記述された種類の混成窓を利用する方が、有利となることがある。循環窓の窓関数は、基本的に伝達関数

【数2】

$$\frac{1}{[1 - \alpha z^{-1}]^2}$$

$$W(z) = \frac{1 - P_2(z/0.9)}{1 - P_2(z/0.4)}, \quad (1)$$

の形の伝達関数となる。ここで、 $P_2(z)$ は10次の伝達関数である。

【0043】LPC予測器は、量子化されていない入力音声に対しフレーム毎にLPC分析を実施することにより得られる。荷重フィルタでは、音声信号にスペクトルのピークがある周波数は弱められ、音声信号にスペクトルの谷がある周波数は強調される。この荷重フィルタを励起の閉ループ量子化に使用すると、符号化雑音のスペクトラムが整形されて、その雑音は、この荷重フィルタがない場合に生成される雑音ほど人の耳には聞こえない

を有する2極フィルタのインパルス応答の鏡像である。極 $\alpha$ が1に近づくにつれて、窓の「尾」はより長くなる。

【0040】後方適応LPC分析のための窓形状は、非常に注意深く選定されなければならない、さもなければ、重要な性能の低下を引き起こしてしまうことになる。 $\alpha = 0.96$ という値は、開ループLPC予測、16 kbps LD-CELP符号器、および多くの低雑音の応用に対して適切であるが、このような値は、不自然で煩わしく聞こえる「水っぽい」歪を生ずることがある。このように、循環窓の有効長さを長くするように $\alpha$ の値を増加させることは、実に有利である。

【0041】循環窓の有効窓長が、窓の始まりから窓関数の値がその最大値の10%の点までの時間間隔として定義される場合、 $\alpha = 0.96$ の循環窓は、3.5ms近辺にピークを有し、有効窓長が約15msである。 $\alpha$ が0.96と0.97の間の値では、通常、10次LPC予測にとって最も高い開ループ予測利得が得られる。しかしながら、 $\alpha$ が0.96の時、水っぽい歪が問題となる。 $\alpha$ を0.99に増加させることにより、窓ピークは約13msに位置を変え、有効窓長は、61msに増加する。このように長くなった窓により、水っぽい歪は完全になくなるが符号化音声の品質は幾分落ちてしまう。従って、 $\alpha = 0.96$ の水っぽい歪もなく、 $\alpha = 0.99$ での音声品質の低下もない $\alpha = 0.985$ は、良い妥協点であることが分かった。 $\alpha = 0.985$ の場合、窓ピークは8.5ms付近で起こり、有効窓長は約40msである。

#### 【0042】知覚的荷重フィルタ

図3、4の説明のための8 kbps LD-CELP構造で使用される知覚加重フィルタは、先に引用したチェンの論文に記述されている16 kbps LD-CELPで使用されたものと好都合にも同じである。これは、

【数3】

ようになる。

【0044】後方LPC分析から得られるLPC予測器は、知覚的荷重フィルタを導き出す目的には使用しないので好都合である。これは、後方LPC分析が、8 kbps LD-CELP符号化音声に基づき、符号化歪のために、LPCスペクトラムが入力音声の真の包含線からはずれてしまうことがあるからである。知覚的荷重フィルタは符号器にしか使われないので、復号器は、符号化過程で使われる知覚的荷重フィルタを知る必要はない。ゆえに図3に示したように、知覚的荷重フィルタの係数を

導くために非量子化入力音声を使うことが可能となる。

#### 【0045】ピッチ予測

ピッチ予測器およびその量子化構造は、図3、4に示した低ビットレート（一般的に8 kbps）LD-CELP符号器と復号器の実施例の主要な部分を占める。それゆえにこれらの構造のピッチに関連した機能の背景と作用は、かなり詳細に説明する。

#### 【0046】背景および概略

図3のピッチ予測器310の一つの実施例では、「音響、音声、信号処理に関するIEEE国際会議会報」p. 243-p. 246（1988年4月）のV. イェンガー（Iyengar）とP. カバル（Kabal）による「低遅延16 kbps音声符号器」に記述されている型の後方適応の3タップ・ピッチ予測器を有利に使用してもよい。しかし、[IEEE地球間通信会議会報] P. 1247-P. 1252（1989年11月）のR. ペティグルー（Pettigrew）とV. クーパーマン（Cuperman）による「16 kbps音声の低遅延ベクトル励起符号化」に記述されている方法に一般的に従って、非音声または無音フレームと出会う度にピッチ・パラメータを再設定することで、3タップ後方適応ピッチ予測器を修正する方が有利である（特にチャネル・エラーに対して頑丈にする点において）。この案によって、女性の音声の知覚品質はいくらか改善されたが、男性の音声に対しては改善がそれほど認められない。さらに、頻繁な再設定をした場合も、この案では、BER =  $10^{-3}$ においてチャネルエラーに対する耐性は、依然として必ずしも十分ではなかった。

【0047】図3におけるピッチ予測器310のもう一つの実施例は、前述のモリヤによる論文に記述されているものに基いている。この実施例では、単一ピッチ・タップが完全に前方に送られ、ピッチ周期は一部は後方に、一部は前方に適応化される。しかしながら、この技術はチャネル誤差に対して過敏である。

【0048】図3の説明のための構造におけるピッチ予測器310の好ましい実施例は、完全な前方適応ピッチ予測に基づくものであることが分かった。

【0049】このような完全前方適応ピッチ予測器の第1の変形において、3タップ・ピッチ予測器は、7ビットに閉ループの量子化されるピッチ時間、および5乃至6ビットに量子化される3タップの閉ループ・ベクトルと共に利用される。このピッチ予測器は、非常に高い予測利得（一般的には、明確な荷重信号領域において5乃至6 dB）を達成する。これは、前述の完全または部分的な後方適応構造よりも、チャネル誤差に対して、より頑丈なものである。しかしながら、20または32の標本の大きさのフレームでは、各フレームに対して、20または32ビットしか利用できない。特に、20標本のフレームの場合、ピッチ予測器に12乃至13のビットを使ってしまうと、励起符号化のためにほんのわずかなビットしか残らなくなる。このようにピッチ予測器のた

めに符号化レートを低減した代案の実施例が望ましいことがしばしばある。

【0050】図3、4の実施例では、小さいフレーム・サイズが使用されるので、近接したフレームのピッチ周期は、かなり相関性が高い。このように、フレーム間予測符号化構造は、ピッチ周期の符号化レートを減少させるのに役立つ。しかしながら、このようなフレーム間方法を設計するに当たって、以下のような課題があった。

1. チャネル誤差に対して、この構造をより頑丈にする方法
2. 無音または無声領域から発声領域に変わる時、ピッチ周期において急激な変化に追従する方法
3. 発声領域において高い予測利得を維持する方法

【0051】これらの課題は、ピッチ周期に対する複雑な4ビット予測符号化構造によって解決される。これは、以下においてさらに十分に説明する。第1の課題を解決するために、チャネル誤差に対するこの方法の耐性（頑丈さ）を高めるために、いくつか処置を講ずる。

【0052】第1に、直前のフレームのピッチ周期から、現在のフレームのピッチ周期を予測するために、単純な1次固定係数予測器を使う。これにより、高次の適応予測器を使うより良好な耐性が得られる。「リーキーな（漏れ易い）」予測器を使うことによって、チャネル誤差の伝播を比較的短い期間に制限することが可能である。

【0053】第2に、ピッチ予測器は、現在のフレームが入力音声の発声領域にあると検出されたときに限って、始動する。つまり、現在のフレームが、発声された音声ではない（たとえば、音節の間または文の間の無声または無音の状態）時は必ず、図3、4の3タップ・ピッチ予測器310は、停止され、リセットされる。フレーム間予測符号化構造では、ピッチ周期で再設定される。これは、チャネル誤差の効果がどれほど長く伝わるかをさらに制限するものとなる。一般的にその効果は、1音節に限定される。

【0054】第3に、本発明の好ましい実施例によるピッチ予測器310では、「通信に関するIEEE国際会議会報」P. 1128-P. 1132（1987年6月）のJ. R. B. デマルカ（De Marca）とN. S. ジェイヤント（Jayanti）による「2値インデックスを多次元量子化器のコードベクトルに割り当てるアルゴリズム」および「エレクトロニクスレター23(12)」P. 654-P. 656（1987年6月）のK. A. ゼガー（Zeger）およびA. ガーショ（Gersho）による「ベクトル量子化におけるゼロ余剰のチャネル符号化」において記述された種類の疑似グレー符号化を用いる。このような疑似グレー符号化は、励起コードブックに使われるだけでなく、3ピッチ予測器のタップのコードブックにも使われる。これにより、チャネル誤差に対する耐性がさらに改善される。

【0055】無声または無音のフレームから発声フレー

ムに変化する時に、ピッチ周期の急激な変化に迅速に追従するという第2の課題を解決するために、2つのステップを踏む。最初のステップは、無声または無音のフレームのように固定のゼロでないバイアス値を使うことである。伝統的にピッチ予測器の出力ピッチ周期は、発声領域を除いて、常にゼロに設定される。これは直感的には自然であるが、このために、ピッチ周期の輪郭は非ゼロ平均のシーケンスとなり、発声領域の始まりでピッチ周期のフレームからフレームへの変化が不必要に大きくなる。無声または無音のフレームのピッチ間隔のように50標本の固定バイアスを使うことによって、発声領域の始まりのこのようなピッチ変化は、減少し、フレーム間予測符号化構造にとって、急激なピッチ変化に迅速に追従することを容易にする。

【0056】ピッチ周期での急激な変化に追従する機能を向上させるために取る第2のステップは、ピッチ周期のフレーム間予測誤差のために4ビット量子化器において大きな外側のレベルを使うことである。-20, -6, -5, -4, . . . , 4, 5, 6, 20に位置する15の量子化器のレベルは、フレーム間差動符号化のために使われ、無声および無音フレームの間の50標本のピッチバイアスの絶対符号化のために、16番目のレベルが使われる。-20から+20の大きな量子化器のレベルは、発声領域の始まりにおける急激なピッチ変化に迅速に追従することを可能にし、-6から+6までのより狭い間隔の内部量子化器のレベルは、従来の7ビット・ピッチ量子化器と同じ精度で、続いて起こるゆっくりとしたピッチ変化に追従することを可能にする。16番目の「絶対」量子化器のレベルにより、現在のフレームは発声されたものではないことを符号器が復号器に伝えることが可能となり、また、従来の予測符号化構造においては一般的である減衰し引きずっている尾を持たずに、ピッチ周期の輪郭を50標本のバイアス値に瞬時に再設定できるようにする。

【0057】50標本ピッチのバイアスの導入および大きな外側の量子化器レベルの利用により、発声領域の初めにおいてわずか2乃至3のフレーム（すなわち、約5乃至12ms）が、符号化ピッチ周期において、実際のピッチ間隔に追従するために、一般的に要求されることが分かった。初期の2乃至3フレームの間、ピッチ予測器はまだ十分な予測利得を得ることができないため、符号化された音声には、より多くの符号化歪（平均2乗誤差の意味において）が含まれる。しかしながら、人の耳は信号変化の領域では符号化歪にあまり敏感ではないため、初期の処理から歪は、ほとんど、あるいは全く知覚されない。

【0058】高い予測利得を達成するという第3の課題を解決するために、本発明によるピッチ・パラメータ量子化の方法および構造は、ピッチ周期の予測符号化において閉ループ量子化を行うように構成する。この構造は

以下のように作用する。まず、ピッチ検出器を用いて、入力音声（開ループ方法）に基づく各フレームのピッチの推定値を得る。現在のフレームが無声または無音の場合、ピッチ検出器は働かなくなり、閉ループでの量子化は必要でなくなる（この場合、16番目の量子化器レベルが送られる）。現在のフレームが音声である場合、ピッチ周期のフレーム間予測誤差が計算される。この予測誤差が6標本より大きい場合、これは、フレーム間予測符号化構造がピッチ周期の大きな変化に追い付こうとしていることを示す。この場合、閉ループ量子化は、大きなピッチ変化に追い付こうとすることを妨害する可能性があるので、行うべきではない。代わりに、15レベルの量子化器を使った直接開ループ量子化が使われる。一方、ピッチ周期のフレーム間予測誤差が6標本より大きくない場合、現在のフレームは、発声された音声区分の安定状態の領域にあることが十分に考えられる。この場合だけ、閉ループ量子化が達成される。ほとんどの音声フレームは、この範疇に入るため、閉ループ量子化は、ほとんどの音声フレームにおいて実際に使われる。

【0059】図3、4のCELP符号器および複合器において使用するために、本発明のピッチ予測器（その量子化構造も含む）の好ましい実施例の基本原則を紹介したので、構造および方法の各構成要素をさらに詳細に説明する。この目的のために、ピッチ周期および3ピッチ予測器タップの量子化構造のブロック/流れ図を図5示す。

#### 【0060】開ループピッチ周期の抽出

第1ステップでは、開ループ法で入力音声からピッチ周期を抽出する。これには、図5の要素510内で、10次LPC逆フィルタ処理を行い、LPC予測残差信号を求める。10次LPC逆フィルタの係数は、各フレームについて、量子化されていない入力音声にLPC分析を行う毎に更新される。（この同じLPC分析は、図3に示す知覚重みづけフィルタ（知覚荷重フィルタ）の係数更新にも使われる。）得られたLPC予測残差信号は、要素515内でピッチ周期を抽出する基となる。

【0061】ピッチ抽出アルゴリズムの設計には、以下の2つの課題がある。

(1) 計算の複雑さが、全8kbps LD-CELP符号器の単一DSP実時間実施が可能な程度以下であること。

(2) 出力ピッチの輪郭が、滑らかであること（すなわち、倍数ピッチ周期は許されない）、そしてピッチ平滑動作のために、余分な遅延は許されない。

(1)の理由は明かである。(2)の理由は、フレーム間のピッチ周期の予測符号化は、ピッチ輪郭が音声の発声領域において滑らかに展開してはじめて有効なためである。

【0062】ピッチ抽出アルゴリズムは、上記のラビナ（Rabiner）およびシェイファ（Schäfer）の参考文献に

記述されている相関ピーク採取処理に基づいている。このようなピーク採取は、DSPの実施に特によく適合する。しかしピッチ周期検出用の直接的相関ピーク採取アルゴリズムと比較し、性能の犠牲無しでの実施効率は、4:1削減と標準相関ピーク採取の組合せによって達成され得る。

【0063】ピッチ周期の効率的な検索は、以下の方法で行われる。開ループLPC予測残差標本は、まず3次楕円フィルタにより、1kHzで低域フィルタされ、ついで4:1削減される。そして、得られた削減信号を使い、5~35の時間的遅れ(20~140ピッチ周期に対応)を伴った相関値が計算され、最大の相関を与える遅れ $\tau$ が求められる。この時間遅延 $\tau$ は4:1削減信号領域における遅延であるので、これに対応し、もとの非削減信号領域で最大相関を与える時間遅延は、 $4\tau-3$ と $4\tau+3$ の間にある。

【0064】もとの時間分解能を得るために、非削減LPC予測残差標本を使って、 $4\tau-3$ と $4\tau+3$ の間の遅れに対する相関値が計算される。ピーク相関を与える遅れは、第1のピッチ周期候補であり、 $p_0$ と示す。このピッチ周期候補は、真のピッチ周期の倍数となることがある。例えば真のピッチ周期が30標本の場合、上記ピッチ周期候補は、30、60、90、ときには120標本になることがある。これは相関ピーク採取法のみならず、多くの他のピッチ検出アルゴリズムに共通した問題である。この問題の共通の解決法は、2~3の後続フレームのピッチ計算値を見て、現フレームの最終的なピッチ計算の前に、平滑化を行うことである。しかしこの方法では、現フレームの最終ピッチ周期を決定する前に、多数のフレームが緩衝されるため、必然的に総システム遅延を増大させることとなる。遅延の増大は、符号化の遅延を小さくしようとする目的と相反することになる。それゆえ、遅延を増大させずに倍数ピッチ周期を除去する方法が考案された。

【0065】この方法は、ピッチ周期の計算が20~32音声標本毎にきわめて頻繁に行われることを利用して行う。ピッチ周期は通常20~140標本となるので、頻繁なピッチ計算は、各音声噴出の先頭で、倍数ピッチ周期が上記の相関ピーク採取過程で出現するより前に、最初に基本のピッチ周期が得られることを意味する。初期時間以降には、相関ピークが先行フレームのピッチ周期の近傍にあるかどうかをチェックすることで基本のピッチ周期を固定化できる。

【0066】 $p_{\blacksquare}$ を先行フレームのピッチ周期とする。上で得られた最初のピッチ周期候補 $p_0$ が、 $p_{\blacksquare}$ の近傍にないならば、時間遅延 $i = p_{\blacksquare} - 6$ ,  $p_{\blacksquare} - 5$ , ...,  $p_{\blacksquare} + 5$ ,  $p_{\blacksquare} + 6$ のための非削減領域内の相関値が、評価される。13の可能な時間遅延の中で、最大の相関値を与える時間遅延が、第2のピッチ周期候補であり、 $p_1$ と示す。

【0067】つぎに2つのピッチ周期候補( $p_0$ ,  $p_1$ )のいずれか1つを最終のピッチ周期計算用に採用し、 $p^*$ と示す。これをするために、群遅延の $p_0$ 標本をもつ単一タップ・ピッチ予測の最適タップ重みが決定される。ついでタップ重みは0~1にクリップされる。第2のピッチ周期候補 $p_1$ にもこの操作が行われる。もし $p_1$ に対するタップ重みが $p_0$ に対するタップ重みの0.4倍より大きいならば、第2の候補 $p_1$ が最終のピッチ計算に使われる。それ以外では、第1の候補 $p_0$ が最終のピッチ計算に使われる。このようなアルゴリズムは遅延を増大させることがない。図5の要素515によって受け持たれるここに述べたアルゴリズムは単純であるが、音声の発声領域における倍数ピッチ周期の除去に、きわめてよく作用する。

【0068】上記の図5の要素515内で得られた開ループ計算ピッチ周期は、図5の4ビット・ピッチ周期量子化器520に渡される。加えて、群遅延の $p_0$ 標本をもつ単一タップ・ピッチ予測のタップ重みが、波形の周期性を示すものとして図5の発声フレーム検出器505に要素515によって供給される。

#### 【0069】発声フレーム検出器

図5の発声フレーム検出器505の目的は、(母音領域に対応する)発声されたフレームの存在を検出することである。こうすることで、それら発声フレームに対し、ピッチ予測をONにし、(無声(unvoiced)、無音、および過渡期のフレームを含む)それ以外のすべての非発声(non-voiced)フレームに対しそれをOFFにすることができる。ここで使われた述語「非発声フレーム(non-voiced frame)」は、発声フレームとして分類されないすべてのフレームを意味する。これは通常、摩擦音に対応する「無声フレーム(unvoiced frame)」とはいくぶん異なっている。上記のラビナー(Rabiner)およびシェイファ(Schafer)の参考文献を見られたい。動機は1音節内へのチャンネル・エラー効果の広がりを制限し、完成度を高めることである。

【0070】非発声ないし無音フレームの間、ピッチ予測をOFFにしてもなんらの顕著な性能低下をもたらさないことに注目されたい。というのも、通常これらのフレームのピッチ予測ゲインは、どのみち0に近いからである。さらに、ときたま非発声や無音フレームを発声フレームとして誤分類しても、無害であることにも注目されたい。というのは、ピッチ予測がすべてのフレームで使用された時にも、CELP符号器は良好に作動するからである。一方、恒常の有声セグメントの途中で、発声フレームを非発声として誤分類すると、音声品質を有意に低下させることになる。それゆえ、われわれの発声フレーム検出器は、この種の誤分類を回避するよう、特別に設計されている。

【0071】発声フレームの検出においては、適応強度しきい、(ピッチ抽出アルゴリズムによって生成され

た) 単一タップ・ピッチ予測のタップ重み、規準化された1次自己相関係数、およびゼロ・クロッシング・レートが(優先順位に従って)利用される。もし各フレームが分離して調査され、そのフレームに基づき単独に即座の有声決定がなされるならば、発声領域に途中でときたま、離れて現れる非発声フレームをなくするのは、一般的にはきわめて困難である。そのようなフレームでピッチ予測をOFFにすると、有意な品質低下を引き起こす。

【0072】この種の誤分類を回避するために、デジタル音声内挿システム(DSI)でよく使われているいわゆる「ハング・オーバ」法が現況での使用のために採用された。ハング・オーバ法は、上で与えた4つの決定パラメータに基づく、予備的な有声/非発声分類を考慮した後処理技術と考えることができる。ハング・オーバを使うと、後続の4つ以上のフレームが予備的に非発声と分類された場合に限り、検出器は公式に非発声フレームと宣告する。この方法は、発声領域の途中における離れた非発声フレームを除去するのに有効である。このような遅延した宣告は、非発声フレームに対してのみ適用される。(宣告は遅延するが、符号器がさらなる緩衝遅延をこうむることはない。)フレームが予備的に有声と分類されると、そのフレームは即座に、公式に有声と宣告され、ハング・オーバ・フレーム・カウンタは0にリセットされる。

【0073】予備的分類は以下の通り作用する。適応強度しきい関数は、標本ごとに指数関数的に、例えば0.9998の減衰係数で減衰する。入力音声標本の大きさがしきいより大きいと、しきいはその値にセット(あるいは更新(refreshed))され、その値から減衰が続ける。標本ごとに現フレーム上で平均化されたしきい関数は、比較の対象として使用される。現フレーム内の入力音声標本のピークの大きさが平均しきいの50%より大きいと、即座に現フレームを有声と宣告する。入力音声標本のピークの大きさが平均しきいの2%より小さいと、現フレームを予備的に非発声と分類し、ハング・オーバ後処理に委ねる。ピークの大きさが平均しきいの2%と50%の間にあるならば、灰色領域にあるとみなし、現フレームを分類するために、次の3つの試験が行われる。

【0074】まず、現フレームの最適単一タップ・ピッチ予測のタップ重みが、0.5より大きいならば、現フレームは有声と宣告する。タップ重みが、0.5より大きくないならば、入力音声の規準化された自己相関係数が、0.4より大きいかどうかを調べる。大きいならば、現フレームを有声と宣告する。大きくないなら、さらにゼロ・クロッシング・レートが0.4より大きいかどうかを調べる。大きいなら、現フレームを有声と宣告する。3つの試験のいずれにも該当しないならば、一時的に現フレームを非発声と分類し、ハング・オーバ後処

理工程を通す。

【0075】この単純な発声フレーム検出器はきわめてよく作動する。実用上、他の8 kbps LD-CELP符号器に比べ、工程が複雑そうに見えるが、この発声フレーム検出器は、実施に無視できるくらいのDSP実時間を要するにすぎない。

【0076】図5において、現フレームが有声と宣告されると、全ての機能ブロックが正常に作動する。一方、発声フレーム検出器が非発声フレームと宣告すると、次の特別な動作が起こる。まず、4ビット・ピッチ周期量子化器(すなわち50標本ピッチ・バイアスの絶対符号化)の16番量子化レベル量子化出力として選択される。つぎに、3ピッチ・タップのVQコードブックから特別の全ゼロ・コードベクタが選択される。すなわち、すべての3ピッチ予測タップがゼロにセットされる。

(こういった特別な制御は、図3の点線で示してある。)第3に、図5の下半分の帰還ループ内のメモリ(遅延ユニット)が、50標本の固定ピッチ・バイアスの値にリセットされる。第4に、ピッチ予測メモリがゼロにリセットされる。加えて、現フレームが発声フレームの後の最初の非発声フレーム(例えば発声領域の後縁など)ならば、チャンネル・エラーを反映する音声符号化内部状態は、都合よくその固有の初期値にリセットされる。全てのこれらの処置は、チャンネル・エラーが、ひとつの発声領域から他に広がるのを制限するためにとられる。じっさいそれらは符号器の粗さを改良し、チャンネル・エラーを防ぐのに役立っている。

【0077】ピッチ周期のフレーム間予測量子化  
ピッチ周期用のフレーム間予測量子化アルゴリズムまたは方式は4ビット・ピッチ周期量子化器520と図5の下半分の予測帰還ループを含む。帰還ループの下側は1入力を比較器560に供給する(他の入力は、50標本に対応したピッチ・バイアスを供給するバイアス源555から入る)遅延要素565と入力をコンパレータ550から受けとり、その出力を加算器545に供給する、標準利得0.96のアンプを含む。加算器、545への他の入力も、バイアス源555から入る。加算器、545の出力は丸め要素、525に供給され、また加算器、570に戻される。加算器、570では外部帰還ループの比較器575からの信号と合算され、遅延要素565の入力となる。図に示したように、丸め要素、525はまた、4ビット・ピッチ周期量子化器への入力を供給する。これら要素の動作を以下に示す。

【0078】4ビット・ピッチ周期量子化器、520は、まず開ループピッチ周期抽出器、515によって生成されたピッチ周期 $p$ から、丸められた予測ピッチ周期 $r$ を減じる。差 $d = p - r$ が、6より大、または-6より小ならば、量子化器の4つの出力レベル、-20、-6、+6、+20の中の差 $d$ に最も近い一つに直接量子化される。このケースでは、上に述べたようにフレーム

間の予測ピッチ量子化器ピッチ周期の大きな変化に追従しようとする。ピッチ周期の閉ループ最適化は行っていないと、さもないと量子化器の変化への追従を妨害することになる。この状況下では、4ビット・ピッチ周期量子化器の出力ポートにおけるスイッチは、上側位置、521に接続されている。差 $d$ の量子化値を $q$ とすると、量子化ピッチ周期は、 $p = r + q$ となる。この量子化ピッチ周期 $p$ は、3ピッチ予測タップの閉ループベクトル量子化において使われる。

【0079】一方、 $d$ が $-6 \sim +6$ 間にあれば、4ビット・ピッチ周期量子化器の出力ポートにおけるスイッチは、下側位置、522に接続され、閉ループ抽出ピッチ周期 $p$ は、さらなる閉ループ最適化を受けることになる。「閉ループ結合ピッチ周期とタップの最適化」と名付けた図5のブロック530の動作を以下に示す。このブロックの2つの出力の内のどちらかが、閉ループ最適化の後、最終的な量子化ピッチ周期 $p$ になる。

【0080】図5のフレーム間のピッチ周期予測用に使われる帰還ループについて以下に示す。ちょっと見ると、構造が普通の予測符号器の構造とかなり違って見える。この差には2つの理由がある。(1) 50標本ピッチ・バイアスが適用され、(2) 予測信号がどんな値でも取れる他の大多数の予測符号化方式と異なり、このピッチ周期は、システムの他の箇所では使われる前に、最も近い整数値に丸められなければならない。

【0081】図5について更に説明する。量子化ピッチ周期は、 $p = r + q$ と表現できる。ここに、フレーム間のピッチ周期予測エラー（例えば上記と異なった値）の量子化値は図5に示すように、 $q = p - r$ として得られる。 $q$ を、 $p$ （予測ピッチ周期の浮動小数点値）に加えた後、加算器、570において、復元されたピッチ周期の浮動小数点値が得られる。 $Z^{-1}$ と名付けた遅延ユニット、565は、要素555によって供給される50標本の固定ピッチ・バイアスを引き、先行フレームの浮動小数点復元ピッチ周期を求めるのに有効である。得られる差は係数、0.94によって減じられ、その結果に50標本のピッチ・バイアスが加えられ、浮動小数点予測ピッチ周期、 $p$ が得られる。この $p$ は、要素525中で最も近い整数に丸められ、丸められた予測ピッチ周期 $r$ となり、これで帰還ループが完結する。

【0082】50標本ピッチ・バイアスの減算、加算が省かれると、図5の下側帰還ループは、従来の予測符号器の帰還ループに還元することに注目されたい。リーケージ係数の目的は、復号ピッチ周期のチャンネル・エラー効果を時間とともに減衰させることである。小さなリーケージ係数は、チャンネル・エラー効果の減衰を速めるが、予測ピッチ周期の、先行フレームのピッチ周期とのずれを大きくする。この点と50標本の必要性を以下に例示する。

【0083】太い男性の声で、先行フレーム100標

本、現フレーム101標本、ピッチ周期が+1標本/フレームの割合で徐々に増加していると想定する。もし50標本ピッチ・バイアスをかけないと、(丸めた)予測ピッチ周期は、 $r = p = 100 \times 0.94 = 94$ 、フレーム間のピッチ周期予測エラーは、 $d = p - r = 101 - 94 = 7$ 、となる。 $d$ は6を超えているので、 $q = 6$ 、に量子化され、量子化ピッチ周期は、 $p = 94 + 6 = 100$ 、となり希望値、101とは異なっている。実際の入力音声のピッチ周期が114標本に達し、4ビット量子化器の出力レベルが+6に替わって+20になるまで、100標本の量子化ピッチ周期を発生し続けるといったように、ピッチ量子化方式が入力音声の遅いピッチ増加にも追従できないのは、何が悪いのか。

【0084】今度は、50標本ピッチ・バイアスを使った例を考える。(丸めた)予測ピッチ周期は、 $r = p = 50 + (100 - 50) \times 0.94 = 97$ 、フレーム間のピッチ周期予測エラーは、 $d = 101 - 97 = 4$ 、となり。これは量子化範囲内であり、予測量子化方式が入力音声の増加に追従できる。

【0085】この例から固定ピッチ・バイアスが望ましいのは、明瞭である。リーケージ係数があまり小さいと、ピッチ周期量子化方式は、入力ピッチ周期の変化を追跡できないこともまた明瞭である。

【0086】ピッチ・バイアスのもう一つの利点は、ピッチ周期量子化方式が発声領域の先頭のピッチ周期の急激な変化により速やかに追従することを可能にすることである。例えば、発声領域の先頭でピッチ周期が90標本とし、ピッチ・バイアスなし（すなわちピッチ0から開始）では、追従するのに6フレームを要するのに対し、50標本ピッチ・バイアスでは追従するのは2フレームにすぎない。(量子化レベル+20が、2回選択されることによる)

【0087】ピッチ予測器のタップの閉ループ量子化 4ビット・ピッチ周期量子化器520が、「キャッチ・アップ・モード」の場合、外側の量子化レベルの中の1つが選択され、その出力にあるスイッチは、上の位置に接続される。この場合、ピッチ周期の調整はこれ以上行われず、量子化されたピッチ周期 $p$ が、3ピッチ予測器のタップの閉ループVQ（ベクトル量子化）において直接使用される。ピッチ予測器タップ・ベクトル量子化器では、3ピッチ予測器タップを量子化し、さらに32または64の所属を有するVQコードブックを用いてそれぞれ5または6ビットに符号化する。

【0088】そのようなベクトル量子化を行う表面的に自然な方法は、3次線形方程式を解いたうえで、歪測度として3つのタップの平均2乗誤差(MSE)を用いて3つのタップを直接ベクトル量子化することにより、3つのタップの荷重の最適集合を最初に計算する。しかし、最終的な目的は、3つのタップのMSE自体を最小にすることではなく、知覚的に加重された符号化雑音を

最小にすることであるから、知覚的に加重された符号化雑音を直に最小化しようとするいわゆる閉ループ量子化を行うことがより良い方法である。ピッチ予測器の量子化および励起信号の量子化は、一括して2段階の連続した近似過程として考えることができるので、加重されたピッチ予測残差のエネルギーを最小にすることにより、低遅延CELP符号化過程全体の総体的歪測度が直に最小化される。直接的な係数のMSE基準と比較して、この閉ループ量子化は、より良いピッチ予測利得を与えるのみならず、総体的な低遅延CELP符号化歪も減少さ

$$P_1(z) = \sum_{i=1}^3 b_{ji} z^{-p+2-i}$$

ただし、 $p$ は、先に決定された量子化されたピッチ周期である。

【0090】フレームの大きさを標本 $L$ 個分とすると、普遍性を欠くことなく、現在のフレームに $k=1$ から $k=L$ まで信号標本にインデックスを付けることができる。正でないインデックスは、前のフレームにある信号

$$f(k) = \sum_{i=1}^3 b_{ji} d(k-p+2-i)$$

ここで、 $L$ 次元の列ベクトルを、

【数6】

$$\mathbf{f} = [f(1), f(2), \dots, f(L)]^T$$

$$\mathbf{f} = \sum_{i=1}^3 b_{ji} \mathbf{d}_i$$

ただし、

$$\mathbf{d}_i = [d(1-p+2-i), d(2-p+2-i), \dots, d(L-p+2-i)]^T \quad (5)$$

を得る。

【0091】留意すべきは、ピッチ周期 $p$ がフレームの大きさ(32標本のフレームの場合)より小さい場合、 $\mathbf{d}_i$  ( $\mathbf{d}_i$ は、数7において太字で記したベクトルと同じものであり、それを本文ではこのように記す)は正のインデックス $k$ を有する成分 $d(k)$ の中の幾つかを持つことである。つまり、それは現在のフレームの幾つかの $d(k)$ 標本を必要とする。しかし、これらの標本は、ピッチ予測器のタップおよび励起コードベクトルの量子化がまだ完了していないので、まだ利用できない。他の従来のCELP符号器における単一タップのピッチ予測器の閉ループ量子化にも同様の問題がある。この問題は、「拡張した適応コードブック」の概念を用いることによって容易に回避することができるが、この概念は、「音響、音声、信号処理に関するIEEE国際会議(IEE Int. Conf. Acoust., Speech, Signal Processing)」会報(1988年4月)のW. B. クリージン (Kleij

せる。しかし、この加重された残差エネルギー基準によるコードブックの探索には、高速の探索方法を用いない限り、計算上さらに高度な複雑さを伴うのが普通である。以下において、8 kbps低遅延CELP符号器で使用される高速探索法の原理を説明する。

【0089】 $b_{j1}$ 、 $b_{j2}$ 、および $b_{j3}$ をピッチ・タップVQコードブックにおける $j$ 番目の所属の3つのピッチ予測器タップであるとすると、対応する3タップのピッチ予測器は、次式の伝達関数を有する。

【数4】

(2)

標本に対応する。 $d(k)$ をLPCフィルタへの励起(即ち、ピッチ合成フィルタへの出力)の $k$ 番目の標本であるとする。すると、 $j$ 番目の候補であるピッチ予測器の $k$ 番目の出力標本は、次の式のように表すことができる。

【数5】

(3)

なる式で定義すると、

【数7】

(4)

$n$ )、D. J. クラシンスキ (Krasinski)、およびR. H. ケトチャム (Ketchum) による「SELPにおける改良された音声品質および効率のベクトル量子化 (Improved speech quality and efficient vector quantization in SELP)」に提案されている。基本的には、前のフレームにおける $d(k)$ の最後の $p$ 個の標本を周期的に繰り返すことにより、現在のフレームに対して $d(k)$ シーケンスが推定される。ただし、 $p$ はピッチ周期である。

【0092】標準的なCELP符号化の処理と同様に、3ピッチ・タップの閉ループ量子化が開始される前に、入力音声の現在のフレームは、知覚的に加重するフィルタに通されて、結果的に加重された音声フレームから加重されたLPCフィルタのゼロ入力応答を減ずる。差信号 $t(k)$ は、ピッチ予測器タップの閉ループ量子化のための目標信号である。 $L$ 次元の目標フレームは、次の式で定義することができる。



【数8】

$$t = [t(1), t(2), \dots, t(L)]^T$$

【0093】縦続接続したLPC合成フィルタおよび知覚的荷重フィルタ（即ち、加重されたLPCフィルタ）のインパルス応答を $h(n)$ とする。 $i \geq j$ の場合は $h_{ij} = h(i-j)$ 、 $i < j$ の場合は $h_{ij} = 0$ によって与えられる $i, j$ 番目の成分を有する $L \times L$ の下半3角行列

$$D_j = \|t - H\eta\|^2 = \|t - H \sum_{i=1}^3 b_{ji} d_i\|^2 = \|t - \sum_{i=1}^3 b_{ji} (H d_i)\|^2, \quad (6)$$

ただし、所与のベクトル $a$ に対し、記号「 $\|a\|^2$ 」は、ユークリッド型ノルムの2乗を示す、即ち $a$ のエネルギーである（これらの $a$ は、すべて太字で記すべきベ

$$c_i = H d_i$$

と定義し、かつ式（6）における項を展開すると、次のようになる。

$$D_j = \|t - \sum_{i=1}^3 b_{ji} c_i\|^2 = \|t\|^2 - 2t^T \sum_{i=1}^3 b_{ji} c_i + \|\sum_{i=1}^3 b_{ji} c_i\|^2 \quad (8)$$

$$= \|t\|^2 - 2 \sum_{i=1}^3 b_{ji} (t^T c_i) + \sum_{i=1}^3 \sum_{m=1}^3 b_{ji} b_{jm} c_i^T c_m \quad (9)$$

$$= E - 2 \sum_{i=1}^3 b_{ji} \phi_i + \sum_{i=1}^3 \sum_{m=1}^3 b_{ji} b_{jm} \psi_{im}, \quad (10)$$

ただし、

$$E = \|t\|^2, \quad (11)$$

$$\phi_i = t^T c_i, \quad (12)$$

かつ、

$$\psi_{im} = c_i^T c_m. \quad (13)$$

式（10）における総和法を展開し、かつ類似の項を縮約すると、式（10）は次のように書くことができる。

を $H$ （この $H$ は、「化学式等を記載した書面」において太字で記したものと同じである）と定義する。この場合、閉ループのピッチ・タップ・コードブック探索に対し、そのピッチ・タップVQコードブックにおける $j$ 番目の候補のピッチ予測器に関係付けられた歪は、次式によって与えられる。

【数9】

クトルである）。

【0094】ここで、

【数10】

(7)

【数11】

【数12】

$$D_j = E - B_j^T C, \quad (14)$$

ただし,

$$B_j^T = [2b_{j1}, 2b_{j2}, 2b_{j3}, -2b_{j1}b_{j2}, -2b_{j2}b_{j3}, -2b_{j3}b_{j1}, -b_{j1}^2, -b_{j2}^2, -b_{j3}^2] \quad (15)$$

かつ,

$$C = [\phi_1, \phi_2, \phi_3, \psi_{12}, \psi_{23}, \psi_{31}, \psi_{11}, \psi_{22}, \psi_{33}]^T. \quad (16)$$

【0095】目標ベクトルのエネルギー項Eはコードブック探索の間は一定であるから、 $D_j$ を最小にすることは、2つの9次元ベクトルBおよびC（ベクトルBおよびCは、「化学式等を記載した書面」において太字で記したものと同一である）の内積

【数13】

$$B_j^T C$$

を最小にすることと同じである。この2つのバージョンの8 kbps低遅延CELP符号器では、3つのピッチ予測器タップの量子化に5または6のビットを使用するので、ピッチ・タップVQコードブックにはピッチ予測器タップの32または64の候補集合がある。以下の説明の便宜上、6ビットのコードブックが使用されるものと仮定する。

【0096】このコードブックにおけるピッチ予測器タップの64の候補集合の各々に対し、それに関係付けられ対応する9次元のベクトル $B_j$ が存在する。64の可能な9次元ベクトル $B_j$ は、都合良く予め計算され記憶されているので、コードブック探索の最中に $B_j$ ベクトルを求める計算の必要はない。また、ベクトル $d_1$ 、 $d_2$ 、および $d_3$ は、互いに少しずつ転位したものであることから、そのような構造が開発された場合、ベクトルCを完全に効率的に計算することができる。実際のコードブック探索では、9次元ベクトルCが一度計算されると、64の記憶されたベクトル $B_j$ との64の内積が計算され、最大の内積を与えるベクトル $B_{j*}$ が特定される。そして、このベクトル $B_{j*}$ の最初の3つの要素に0.5を乗ずることによって、3つの量子化された予測器タップが得られる。1フレームごとに、6ビットのインデックス $j_*$ が、出力ビット・ストリーム・マルチプレクサに渡される。

【0097】現在のフレームが発声フレームでない時にピッチ予測器を完全に停止させることができるように、ピッチ・タップVQコードブックにゼロ・コードベクトルを挿入してある。その他の31または63のピッチ・タップ・コードベクトルが、コードブック設計アルゴリズムを用いて、閉ループで仕込まれる。この時のコードブック設計アルゴリズムは、委員会28の通信に関する

IEEE会報 (IEEE Trans. Comm., Comm. 28) p.84-p.95 (1980年1月) のY. リンデ (Linde)、A. バゾ (Buso)、およびR. M. グレイ (Gray) による「ベクトル量子化器設計のためのアルゴリズム (An algorithm for vector quantizer design)」において説明された種類のものである。発声フレーム検出器が非発声フレームを宣言すると、如何なる場合も、ピッチ周期を50標本分のバイアス値に設定し直すだけでなく、すべてゼロのコードベクトルをピッチ・タップVQ出力として選択する。つまり、3つのピッチ・タップがすべてゼロに量子化される。従って、4ビットのピッチ周期インデックス、および5または6ビットのピッチ・タップ・インデックスの両方を非発声フレームを示すものとして使用することができる。発声された領域の中央で発声されたフレームを誤って非発声として復号すると、一般に極めて厳しい音声品質の劣化を招くが、この種のエラーは、可能ならば避けるべきである。従って、復号器では、4ビットのピッチ周期インデックスおよび5または6ビットのピッチ・タップ・インデックスの両方が、現在のフレームが非発声フレームであることを示す場合に限って、現在のフレームを非発声のものであると宣言する。両インデックスを非発声フレームの指示子として用いることにより、発声フレームを非発声のものとする復号エラーを防ぐタイプの冗長性が与えられる。

【0098】これまで、図5において「3つのピッチ・タップの閉ループVQ (ベクトル量子化)」と記されたブロック530によって代表される機能を、フレーム間のピッチ周期の予測誤差が6標本を超える大きさである場合に対して説明してきた。次に、そのようなピッチ周期の予測誤差の大きさが6標本に等しいか、それ以下である場合を説明する。この場合、閉ループの意味でより良いピッチ周期を発見できるという見込みをもってピッチ周期のさらに細かな調節をする機会がある。従って、4ビット・ピッチ量子化器の出力にあるスイッチ523は、ピッチ周期およびタップの閉ループ連帯最適化を許すために下の位置522に位置決めされる。

【0099】理想的に言えば、探索の際に、ピッチ量子化器の13のレベル（-6から6まで）と3タップVQコードブックの32または64のコードベクトルとの可

能なすべての組み合わせを通して、最良の閉ループ量子化結果が得られることである。しかし、そのような徹底的な連帯探索の計算的複雑さは、実時間の実施には過度であることもある。従って、比較的簡単な改善の方法を求める方が有利となる。

【0100】本発明の応用として使用し得るような方法の第1の実施例には、従来の（単一タップのピッチ予測器の公式化に基づく）CELP符号器と同じ方法を用いてピッチ周期の閉ループ最適化を最初に行うことをが必然的に含まれる。結果的に閉ループ最適化されたピッチ周期が $p^*$ であったとすると、3つの別々の閉ループピッチ・タップ・コードブック探索が、前述の高速探索方法により、3つの可能なピッチ周期 $p^*-1$ 、 $p^*$ 、および $p^*+1$ （勿論、 $[r-6, r+6]$ という量子化器の範囲制限による）について行われる。この方法は、極めて高いピッチ予測利得が得られるが、用途によっては許容できない複雑さが依然としてある。

【0101】計算上の複雑さを少なくする第2の好ましい方法では、ピッチ周期の閉ループ量子化は省略するが、3ピッチ・タップの閉ループ量子化の実行時は5つの候補ピッチ周期が許される。5つの候補ピッチ周期は、 $p_- - 2$ 、 $p_- - 1$ 、 $p_-$ 、 $p_+ + 1$ 、および $p_+ + 2$ （同様に $[r-6, r+6]$ の範囲制限に従う）であった。ただし、 $p_-$ は、閉ループ・ピッチ抽出アルゴリズムによって得られたピッチ周期であった。これは、ピッチ量子化器の範囲を狭くして（ピッチ周期の候補を13ではなく5にして）閉ループの要領でピッチ周期およびピッチ・タップを連带的に量子化することに相当した。この比較的簡単な方法によって得た予測利得は、第1の方法のそれに匹敵した。

#### 【0102】ピッチ予測器の性能

以上説明した複雑なフレーム間ピッチ・パラメータ量子化方式によって、7ビットのピッチ周期および5または6ビットのピッチ・タップを有する最初の方法とほぼ同じピッチ予測利得（知覚的に加重された信号範囲において5~6 dB）を達成することができた。さらに、我々が普通に聞いたところによれば、雑音がちのチャンネル状態の下では、従来の7ビットのピッチ量子化器または本発明の4ビットのフレーム間予測量子化器の何れを用い

た場合も、全く匹敵する音声品質が得られた。換言すれば、ピッチ予測利得もチャネル・エラーに対する強度も妥協することなく、ピッチ周期の符号化率を7 bit/フレームから4 bit/フレームまで下げたことになる。この3ビットの節約は、一見、重要なことではないかも知れないが、この小さなフレーム・サイズにより、この節約は、全ビットレートの10乃至15%程度（750~1200 bps）に相当する。これらの3ビットを励起コードベクトルの符号化に割り当てた後では符号化された音声の知覚品質が著しく改善されることを発見した。

#### 【0103】利得の適応

励起利得適応方法は、16 kbps低遅延CELPアルゴリズムの場合と本質的に同じである。「音響、音声、信号処理に関するIEEE国際会議会報」p.181-p.184（1990年4月）のJ. H. チェン（Chen）による「一方向の遅延が2ms以下の高品質16 kbps低遅延CELP音声符号化（High-quality 16 kbit/s low-delay CELP speech coding with a one-way delay less than 2ms）」参照。励起利得は、対数利得変域で動作させた10次線形予測器によって後方適応化される。この10次の対数利得予測器の係数は、フレームごとに、倍率調整された励起ベクトルの前の対数利得に対し後方適応LPC分析を行うことによって、更新される。

#### 【0104】励起符号化

次の表1に、フレーム・サイズ、励起ベクトルの次元、ならびに本発明の実施例による8 kbps低遅延CELP符号器の2つのバージョンおよび6.4 kbps低遅延CELP（以下、「LD-CELP」と記す）符号器を示す。フレーム・サイズが20標本の8 kbps版の符号器は、各フレームに1つの励起ベクトルを収容している。一方、32標本/フレームの符号器は、各フレームに2つの励起ベクトルを持つ。6.4 kbpsのLD-CELP符号器は、単に32標本/フレームの符号器のフレーム・サイズおよびベクトルの次元を大きくし、その他はすべて同じに維持することによって得られる。3つのすべての符号器において、各励起ベクトルに対し、励起形状コードブックに7ビット、強度コードブックに3ビット、そして符号に1ビットを使用する。

【表1】

ビット速度	8 kb/s	8 kb/s	6.4 kb/s
フレーム・サイズ (ms)	2.5	4	5
フレーム・サイズ (標本数)	20	32	40
ベクトルの次元	20	16	20
ベクトル数/フレーム	1	2	2
ピッチ周期 (ビット)	4	4	4
ピッチ・タップ (ビット)	5	6	6
励起符号 (ビット)	1	1×2	1×2
励起強度 (ビット)	3	3×2	3×2
励起形状 (ビット)	7	7×2	7×2
全ビット数/フレーム	20	32	32

第1表 低遅延CELP符号器のパラメータとビット割当

【0105】これらの実施例で使用する励起コードブック探索の手順および方法は、16 kbps LD-CELPのコードブック探索とは幾分異なる。8 kbpsの方がベクトルの次元および利得コードブックの大きさが大きいので、引用したチェンの論文に記述された比較的前の16 kbps LD-DELP方法で使用されたものと同じコードブック探索手順を使用すると、計算上極めて複雑となり、例えば、単一の80 nsのAT&T DSP 32 Cチップのような特定のハードウェア上に全二重符号器を実施することは不可能になる。従って、コードブック探索の複雑さを軽減する方が、有利である。

【0106】8 kbpsと16 kbpsのLD-CELP符号器の間のコードブック探索方法には、2つの主な相違がある。第1に、複雑さを軽減するためには、16 kbpsの場合のように励起の形状および利得を連帯して最適化するより、8 kbpsで形状そして利得というように順に

最適化の方が有利である。第2に、16 kbpsの符号器がフィルタ処理された形状コードベクトルのエネルギー（時として、「コードブック・エネルギー」と呼ばれることがある）を直に計算するのに対し、8 kbpsの符号器では、はるかに高速な新奇な方法を使用する。以下において、まずコードブック探索手順を説明し、続いて、コードブック・エネルギーを計算する第1の方法を説明する。

#### 【0107】励起コードブックの探索手順

励起コードブック探索を開始する前に、3タップ・ピッチ予測器の量子化のために、目標フレームからピッチ予測器の貢献分を引く。結果として、励起ベクトル量子化のための目標ベクトルを得る。これは、次のように算出される。

【数14】

$$\mathbf{x}(n) = \mathbf{t} - \sum_{i=1}^3 \mathbf{b}_i \mathbf{c}_i, \quad (17)$$

【0108】ただし、この式の右辺の記号は、すべて前記の「ピッチ予測器のタップの閉ループ量子化」と題する節において定義したものである。以降の説明を明確にするために、ここでは、ベクトルの時間インデックス  $n$  を励起目標ベクトル  $\mathbf{x}(n)$  に追加した。

【0109】20 標本／フレーム版の 8 kbps L D - C E L P 符号器の場合、励起ベクトルの次元は、フレーム・サイズと同じであり、励起コードブック探索に励起目標ベクトル  $\mathbf{x}(n)$  を直接使用することができる。これに対して、(第1表の2列目および3列目のように) 各フレームに1つ以上の励起ベクトルが入っている場合、励起目標ベクトルの計算は、さらに複雑になる。この場合、まず式(17)を用いて励起目標フレームを計算する。すると、第1の励起目標ベクトルは、励起目標フレームの対応する部分と標本毎に等しい。しかし、第2のベクトルからは、 $m$  番目の励起目標ベクトルを計算するとき、励起ベクトル1から励起ベクトル  $(n-1)$  のために加重された L P C フィルタのゼロ入力応答を励起目標フレームから引かなければならない。これを行うのは、加重された L P C フィルタの記憶の影響を分離する

$$\mathbf{D} = \|\mathbf{x}(n) - \mathbf{H}\mathbf{e}(n)\|^2 = \sigma^2(n) \|\hat{\mathbf{x}}(n) - \mathbf{g}_i \mathbf{H}\mathbf{y}_j\|^2, \quad (19)$$

ただし、 $\hat{\mathbf{x}}(n) = \mathbf{x}(n)/\sigma(n)$  は、利得正規化した励起 V Q 目標ベクトルである。

ここでも、加重された L P C フィルタのインパルス応答の標本によって占められた副対角要素 (subdiagonals) を有する下半三角行列を表すのに、便宜上、記号  $\mathbf{H}$  を用いる。この行列は、その大きさが  $L \times L$  ではなく  $K \times K$  である点を除くと、段落 93 の  $\mathbf{H}$  行列と全く同じ形式で

ためである。これにより、加重された L P C フィルタのインパルス応答による畳み込みによって、励起コードベクトルのフィルタ処理を行うことができる。さらに好都合となるように、 $n$  番目の励起ベクトルに対する最後の目標ベクトルを表すのに記号  $\mathbf{x}(n)$  を依然として使用する。

【0110】7ビット形状コードブックにおける  $j$  番目のコードベクトルを  $\mathbf{y}_j$  とし、後方利得適応方式によって評価された励起利得を  $\sigma(n)$  とする。3ビットの強度コードブックおよび1つの符号ビットを組み合わせ、(正負の両利得に関する) 4ビットの「利得コードブック」を得ることができる。この4ビットの利得コードブックにおける  $i$  番目の利得レベルを  $g_i$  とする。励起コードブック・インデックスの対  $(i, j)$  に対応する倍率調整された励起ベクトル  $\mathbf{e}(n)$  は、次のように表される。

【数15】

$$\mathbf{e}(n) = \sigma(n) \mathbf{g}_i \mathbf{y}_j \quad (18)$$

【0111】インデックスの対  $(i, j)$  に対応する歪は、次式によって与えられる。

【数16】

ある。ここで、 $K$  は、励起ベクトルの次元 ( $K \leq L$ 、かつ  $L/K = \text{正の整数}$ ) である。式(19)の項を展開すると、次式を得る。

【数17】

$$\mathbf{D} = \sigma^2(n) \left[ \|\hat{\mathbf{x}}(n)\|^2 - 2\mathbf{g}_i \hat{\mathbf{x}}^T(n) \mathbf{H}\mathbf{y}_j + g_i^2 \|\mathbf{H}\mathbf{y}_j\|^2 \right]. \quad (20)$$

【0112】項  $\|\hat{\mathbf{x}}(n)\|^2$  および  $\sigma^2(n)$  の値はコードブック探索の間は固定されるので、 $\mathbf{D}$  を最小にす

ることは、次の式を最小かすることに等しい。

【数18】

$$\hat{\mathbf{D}} = -2\mathbf{g}_i \mathbf{p}^T(n) \mathbf{y}_j + g_i^2 \mathbf{E}_j, \quad (21)$$

ただし、

$$\mathbf{p}(n) = \mathbf{H}^T \hat{\mathbf{x}}(n), \quad (22)$$

かつ、

$$\mathbf{E}_j \|\mathbf{H}\mathbf{y}_j\|^2. \quad (23)$$

【0113】 $E_j$ は、実際には $j$ 番目のフィルタ処理された形状コードベクトルのエネルギーであり、VQ目標ベクトル $\hat{x}^*(n)$ には依存しないことに注意を要する。また、形状コードベクトル $y_j$ は固定であり、行列 $H$ のみがLPCフィルタおよび荷重フィルタ（これらは、各フレームにわたって固定されている）に依存する点に注意を要する。都合の良いことに、 $E_j$ も各フレームにわたって固定されている。従って、各フレームに1つ以上の励起ベクトルが含まれる限り、各フレームの最初に128の可能なエネルギー項 $E_j$  ( $j=0, 1, \dots, 127$ )を計算して格納しておき、これら

$$\hat{D} = -2g_i P_j + g_i^2 E_j \quad (25)$$

【0115】16 kbps LD-CELPのコードブック探査では、式(25)の $\hat{D}$ を最小にするインデックスの組み合わせを見つけるために、2つのインデックス $i$ および $j$ の可能なすべての組み合わせが調べられる。しかし、8 kbps符号器の利得コードブックの大きさは16 kbps符号器のその2倍の大きさであるから、そのような形状と利得の連帯的最適化を行うと、探査の複雑さがかなり増大する。従って、最初に最良の形状コードベクトルを捜し、次に既に選ばれた形状コードベクトルに対して最良の利得レベルを決定することによって、複雑さを軽減するために別の次善の方法を使用する方が有利である。事実、この方法は、他のほとんどの通常の前方向適応CELP符号器によって使用されている。この周知の方法においては、最初に、利得 $g_j$ は「流動的」で如

$$\hat{D} = -\frac{P_j^2}{E_j} \quad (27)$$

を得る。従って、形状コードブックの最良のインデックスは、 $P_j^2/E_j$ を最大にするインデックスを見つけることによって決定される。形状コードブックの選択された最良のインデックス $j$ が与えられると、4ビット利得コードブックを用いて最適利得 $g_j^*$ を直に量子化することによって、対応する最良の利得インデックスを発見することができる。利得の量子化は、形状コードブックの探査ループから外れるので、探査の複雑さが大いに軽減される。一度、最良の形状コードブック・インデックスおよびそれに対応する利得コードブック・インデックスが特定されると、それら2つのインデックスを連結して、単一の11ビットの符号語を形成し、この符号語を出力ビットストリーム・マルチプレクサに渡す。

【0116】128個のフィルタ処理された（即ち、畳み込まれた）コードベクトル $H y_j$  ( $j=0, 1, 2, \dots, 127$ )がすべて同じユークリッド型ノルムを持つ場合、前記の順次最適化の原則によって、連帯的最適化探査方法と同一の出力インデックス $i$ および $j$ が得られる。実際には、行列 $H$ は時間的に変化するの

のエネルギー項をそのフレームのすべてのベクトルに繰り返し使用することにより、計算を節約することができる。

【0114】次のように定義すると、

【数19】

$$P_j = p^T(n) y_j, \quad (24)$$

$\hat{D}$ の式は、さらに次のように簡単化することができる。

【数20】

$$(25)$$

何なる値もとることができると仮定する（即ち、量子化されていない利得を想定する）。従って、

【数21】

$$\partial \hat{D} / \partial g_i = 0$$

と設定することにより、最適な量子化されていない励起利得を

【数22】

$$g_i^* = \frac{P_j}{E_j} \quad (26)$$

として得ることができる。式(25)において $g_i = g_i^*$ を代入して、

【数23】

$$(27)$$

で、 $H y_j$ ベクトルは、一般に同じノルムを持たない。この条件に対する精密な近似は、128の固定された $y_j$ コードベクトルが同じノルムを持つことを要求することにより、達成することができる。従って、励起形状コードブックの閉ループ設計の後に、コードベクトルの全部が単位ユークリッド型ノルムを持つように、コードブックを正規化する。このような正規化手順は、符号化性能の目立つ劣化の原因とはならない。

【0117】従来のCELP符号器において連帯的最適化の方法ではなく順次最適化の方法を用いると、励起利得の量子化が十分な解を持つ限り目立つ性能上の劣化がないことに他の研究者は注目してきた。比較的以前の16 kbps LD-CELPにおいて、2ビットの強度コードブックに関して、順次最適化を用いると著しい劣化が有り得ることが分かった。従って、その場合は、形状および利得の連帯的最適化が本当に必要である。一方、8 kbps LD-CELP符号器では、利得の量子化において一層の解像度を与える3ビットの強度コードブックに関して、順次最適化による相対的な劣化は小さいので本

質的に無視できることが分かった。

#### 【0118】コードブック・エネルギーの計算

既に概観した励起コードブック探索の原理に関して、 $j = 0, 1, 2, \dots, 127$ に対するエネルギー $E_j$ の計算を説明する。 $E_j$ の直接計算には、行列とベクトルの乗算 $H y_j$ 、およびこれに続くその結果の $K$ 次元ベクトルのエネルギー計算が伴う。128個すべての $E_j$ 項に必要な乗法演算の総数は、 $128 \times [K(K+1)/2 + K]$ である。従って、計算上の複雑さは、励起ベクトルの次元 $K$ と共に本質的に徐々に増大する。

【0119】16 kbps LD-CELP符号器では、ベクトルの次元は、非常に低い（僅か5標本）ので、これらのエネルギー項は直接計算することができる。しかし、8 kbps以下のLD-CELP符号器では、使用される最低のベクトル次元は、16（第1表参照）である。このようなベクトル次元の場合、コードブック・エネルギーの直接計算だけで、AT&TのDSP32Cチップ上で実施するには毎秒約4.8百万命令（MIPS）がかかる。符号器および復号器におけるコードブック探索およびその他の仕事を考慮すると、全二重符号器に必要な対応する全DSP処理能力は、そのような80nsのDSP32Cで利用できる12.5MIPSを超える可能性がある。従って、コードブック・エネルギーの計算の複雑さを軽減することが望ましい。

【0120】CELP符号化の文献ににおいて、コードブック探索およびコードブック・エネルギーの計算の複雑さを軽減するべく、いくつかの技法が提案されてきた。（これらの技法の包括的な概観のためには、「音響、音声、信号処理に関するIEEE会報（IEEE Trans. Acoust., Speech, Signal Processing）」ASSP-38(8) p.1330-p.1342（1990年8月）のW. B. クリージン（Kleijn）、D. J. クラシンスキ（Krasinski）、およびR. H. ケトチャム（Ketchum）による「CELP音声符号化アルゴリズムのための高速な方法（Fast method for the CELP speech coding algorithm）」がある。）しかし、これらの技法の多くは、複雑さの軽減を実現するために励起形状コードブックに組み込まれた特

殊な構造に依存するものである。LD-CELPの場合は閉ループで仕込まれた励起形状コードブックを用いることが極めて重要であり、さらにこのコードブックは、反復性のアルゴリズムによって仕込まれるため特殊な構造を持たないという理由から、それらの方法は、LD-CELPには明らかに不適當である。（注意を要することであるが、後方適応LPC予測器は、低遅延符号化により適しているが、音声波形における冗長性の除去においては通常のCELP符号器の前方適応LPC予測器ほど効率的でないこともある。結論として、励起の符号化は、所望の精度まで励起の量子化をするという比較的大きな負荷を持つので、LD-CELP符号器の全体的な性能にとって、良く仕込まれたコードブックが決定的となり得る。）

【0121】構造化されていないコードブックに利用可能な複雑度軽減方法は僅かしかない。それらの大半は、複雑度を軽減するには非効率的であったり、莫大なメモリを必要としたりする。1つの例外は、「音響、音声、信号処理に関するIEEE国際会議（IEEE Int. Conf. Acoust., Speech, Signal Processing）」会報p.2375-p.2379（1986年）のI. M. トランコソ（Trancoso）およびB. S. アタル（Atal）による「確率的符号器において最適なイノベーションを発見する効率的な手順（Efficient procedures for finding the optimum innovation in stochastic coders）」に説明されている自己相関法であり、この方法は、必要なメモリはほどほどに増加するだけで、計算上も実に効率的である。

【0122】この自己相関法は、次のように作用する。ベクトルの次元 $K$ が十分大きい場合、加重されたLPCフィルタのインパルス応答シーケンス $\{h(k)\}$ は、 $k$ が $K$ に近づくにつれて、ほぼゼロに減衰するものと仮定する。（この仮定は、 $K$ が40またはそれ以上の場合、通常のCELP符号器に対して大体成立する。）個のように仮定すると、エネルギー項 $E_j$ は、次のように近似できる。

【数24】

$$E_j = \|H y_j\|^2 \approx \mu_0 v_{j0} + 2 \sum_{i=1}^{K-1} \mu_i v_{ji} = \hat{E}_j, \quad (28)$$

ただし、 $\mu_i$ は、インパルス応答ベクトル $[h(0), h(1), \dots, h(K-1)]^T$ の $i$ 番目の自己相関係数であり、次の式で算出される。

【数25】

$$\mu_i = \sum_{k=0}^{K-1-i} h(k) h(k+i), \quad (29)$$

$v_{ji}$ は、 $j$ 番目の形状コードベクトル $y_j$ の $i$ 番目の自

$$v_j = [v_{j0}, 2v_{j1}, 2v_{j2}, \dots, 2v_{j,K-1}]^T, \quad j = 0, 1, 2, \dots, 127, \quad (31)$$

己相関係数であり、

【数26】

$$v_{ji} = \sum_{k=0}^{K-1-i} y_j(k) y_j(k+i), \quad (30)$$

によって算出される。ただし、 $y_j(k)$ は、 $y_j$ の $k$ 番目の成分である。従って、128個の $K$ 次元ベクトル

【数27】

を再計算して記憶しておく、実際の符号化の最中には、まず  $K(K+1)/2$  回の乗算を用いて、 $K$  次元ベ

$$\mathbf{m} = [\mu_0, \mu_1, \mu_2, \dots, \mu_{K-1}]^T, \quad (32)$$

を計算し、次に、 $128 \times K$  回の乗算を用いて、 $128$  個の近似されたコードブック・エネルギー項を

$$\hat{E}_j = \mathbf{m}^T \mathbf{v}_j, \quad j = 0, 1, 2, \dots, 127. \quad (33)$$

として計算することができる。この方法における乗算の総数は、僅か  $128[K + K(K+1)/256]$  であり、ベクトルの次元  $K$  と共に、およそ直線的に（直接計算の場合に 2 次的であるのに対し）増加する。これに払う代償は、コードブックのメモリが 2 倍必要になることである。2 つのテーブルを記憶する必要があるため、1 つは、形状コードブック自体のもので、他の 1 つは、 $128$  個の自己相関ベクトル  $\mathbf{v}_j$  ( $j = 0, 1, \dots$ ) のものである。

【0123】このメモリ必要量の増加は、一般の  $8 \text{ kbps LD-CELP}$  の実施においては許容できるものである。従って、この方法を用いて、コードブック・エネルギーの計算の複雑さを実例のレベルである  $4.8 \text{ MIPS}$  から  $0.61 \text{ MIPS}$  へと減少させることができる。この方法を適用した後は、単一の  $\text{AT\&T DSP32C}$  チップ上で全二重符号器を実施することができる。この方法は、一般の実施例において大抵の場合は良く役立つが、時として、エネルギー項の近似が十分でないこともある。このような場合には、コードブック探索に誤りが起こる可能性があり、不適切な候補の形状コードベクトルを採用することもある。最終的な結果として、出力の符号化された音声に、時々ではないが希に劣化した音節が現れる。この問題の原因は、僅か  $16$  または  $20$  というベクトルの次元  $K$  では、 $k$  が  $K$  に近付くと共に  $h(k)$  がほぼゼロまで減衰するには、すべての場合において十分に大きいとは限らないことによる。

【0124】この問題に対処するために、コードブック・エネルギーを計算する新たな方法が考案された。その基本概念は、インパルス応答シーケンスをすべて制御することは不可能かも知れないが、 $128$  個の固定された形状コードベクトル  $\mathbf{y}_j$  ( $j = 0, 1, 2, \dots, 127$ ) の各々に関する直感的な知識は確かに存在する——従って、それらは前もって処理することができる、というものである。この方法を理解するために、

【数30】

$$E_j = \|\mathbf{H}\mathbf{y}_j\|^2$$

を考える。 $K$  次元ベクトル  $\mathbf{H}\mathbf{y}_j$  は、基本的には、2 つの  $K$  次元ベクトル  $\mathbf{y}_j$  と  $\mathbf{h} = [h(0), h(1), h(2), \dots, h(K-1)]^T$  との間の畳み込み演算の最初の  $K$  個の出力標本である。畳み込みは可換性の演算であるから、 $E_j = \|\mathbf{H}\mathbf{y}_j\|^2$  と書かずに、

クトル

【数28】

$$\mathbf{m} = [\mu_0, \mu_1, \mu_2, \dots, \mu_{K-1}]^T, \quad (32)$$

【数29】

$$\hat{E}_j = \mathbf{m}^T \mathbf{v}_j, \quad j = 0, 1, 2, \dots, 127. \quad (33)$$

【数31】

$$E_j = \|\mathbf{Y}_j \mathbf{h}\|^2, \quad (34)$$

と表すことができる。ただし、 $\mathbf{Y}_j$  は、 $m \geq n$  のとき  $y_j(m-n)$  に等しく、 $m < n$  のとき 0 に等しい  $m \times n$  番目の要素を有する  $K \times K$  の下半 3 角行列である。これは、 $h$  および  $128$  個の可能な「インパルス応答ベクトル」 $\mathbf{y}_j$  ( $j = 0, 1, 2, \dots, 127$ ) のコードベクトルを持つことに等しい。従って、自己相関法（式（28）の右辺）は、ベクトルの終わりに向かって小さな成分を有するような  $\mathbf{y}_j$  ベクトルに対し、エネルギー項の極めて良好な近似を生成する。一方、ベクトルの先頭の近くに比較的小さい成分を有し、終わりに向かって徐々に大きな成分を有するような  $\mathbf{y}_j$  ベクトルは、実際のインパルス応答ベクトル  $h$  がどうであれ、常に劣等なエネルギー近似を生じる傾向がある。これらの「問題を起こす」コードベクトルは、「危険な」コードベクトルと称する。秘訣は、これらの危険なコードベクトルをコードブックから識別し、正確な計算によって、それに対応するエネルギー項を得ることである。

【0125】危険なコードベクトルをその他から区別するための適切な基準を見つけることは、容易な仕事ではない。なぜなら、エネルギー近似誤差が、時間で変化するインパルス応答ベクトル  $h$  の形に依存するからである。次の統計的な方法は採用して好都合であった。エネルギー近似誤差 (dB) は、

【数32】

$$\Delta_j = 10 \log_{10} \frac{\hat{E}_j}{E_j}, \quad (35)$$

と定義される。ただし、 $\hat{E}_j$  および  $E_j$  は、式（28）で定義されている。

【0126】形状コードベクトル  $\mathbf{y}_j$  が与えられているとすると、それに対応するエネルギー近似誤差  $\Delta_j$  は、インパルス応答ベクトル  $h$  にのみ依存する。実際の  $\text{LD-CELP}$  符号化では、ベクトル  $h$  は、フレームごとに変わるので、 $\Delta_j$  もフレームごとに変わる。従って、 $\Delta_j$  は、確率変数として処理され、その平均および標準偏差は次のように評価される。 $8 \text{ kbps LD-CELP}$  符号器を用いて非常に大きな音声ファイル（仕込用の集合）を符号化し、計算過程で  $\Delta_j$  ( $j = 0, 1, 2, \dots, 127$ ) を各フレームに対して計算し、また各



j に対しフレーム全体にわたって  $\Delta_j$  および  $\Delta_j^2(n)$  の総和を積算する。仕込用の集合に N フレームあるものと仮定し、さらに  $\Delta_j(n)$  を n 番目のフレームにある  $\Delta_j$  の値とする。すると、仕込用の集合を符号化した後は、 $\Delta_j$  の平均（または期待値）が、

【数 3 3】

$$\sigma_{\Delta_j} = \frac{1}{N} \sum_{n=1}^N \Delta_j^2(n) - \left[ E[\Delta_j] \right]^2 \quad (37)$$

【0 1 2 7】  $\Delta_j$  の平均が利用できるようになると、自己相関法のエネルギー近似誤差を小さくすることができる。自己相関法によって生成した近似されたコードブック・エネルギー項  $E_j^*$  は、常に真のエネルギー  $E_j$  の過大な推定値となることが分かる。（つまり、 $\Delta_j \geq 0$  である。）換言すれば、 $E_j^*$  は、 $E_j$  の偏った推定値である。10 の  $-E[\Delta_j]$  / 10 乗を  $E_j^*$  に乗じる（これは、 $E_j^*$  の dB 値から  $E[\Delta_j]$  を引くことに相当する）と、結果的に得られる値は、 $E_j$  の偏っていない推定値となり、エネルギー近似誤差が減少する。

【0 1 2 8】 所与の  $\Delta_j$  が小さい標準偏差を有する場合、それは予測可能性が高く、その平均値は、如何なる特定のフレームにおいても、その実際の値に対する最良の推定値として使用することができる。これに対して、 $\Delta_j$  が比較的大きな標準偏差を持つ場合、それは一段と予測可能性が低く、その平均値を推定値として用いると、やはり大きな平均推定値誤差が得られる。従って、 $\Delta_j$  の大きな標準偏差を有するようなコードベクトルは、「問題を起こす」と考えられる。なぜなら、仮に  $\Delta_j$  の平均値をもってしても、それらの危険なコードベクトルは依然として大きなエネルギー近似誤差を生じるからである。従って、危険なコードベクトルを識別するための基準として  $\Delta_j$  の平均標準偏差を用いるのは、意味のあることである。

$$\hat{E}_j = \mu_0 v_{j0} + 2 \sum_{i=1}^{K-1} \mu_i v_{ji} \text{ for } j = M+1, M+2, \dots, 127.$$

の予備的な推定値を計算する。

3.  $E_j^*$  の推定値の偏りを修正し、最終的なエネルギー

$$E_j^* = \hat{E}_j \left[ 10^{-E[\Delta_j]/10} \right] \text{ for } j = M+1, M+2, \dots, 127.$$

を計算する。

【数 3 7】

$$10^{-E[\Delta_j]/10}$$

の 1 2 8 - M 個の項は、予め計算し、テーブルに記憶して、計算を節約することができる。

【0 1 3 1】 1 2 8 のコードブック・サイズに対して M が 1 0 と小さい場合、音節の劣化という希な事象もすべて完全に回避されることが分かった。説明用の実施例においては、M = 1 6、即ちコードブック・サイズの 1 /

$$E[\Delta_j] = \frac{1}{N} \sum_{n=1}^N \Delta_j(n), \quad (36)$$

として容易に得られる。 $\Delta_j$  の標準偏差は、次の式で与えられる。

【数 3 4】

【0 1 2 9】 これらの危険なコードベクトルが特定されても、それらがコードブック全体に分散している場合、コードブックを進んで行きながら、それらを特別に処理しようとするときには、かなりの間接的な負担がある。従って、それらをすべてコードブックの始めに配置することが望ましい。これを行うために、 $\Delta_j$  の標準偏差に基づき、かつ  $\Delta_j$  の標準偏差がインデックス j の増加と共に減少するように励起形状コードベクトルを並べ替えて、ソート（分類）を行う。 $\Delta_j$  の平均値も、相応に並べ替える。図 6 および 7 は、分類・並べ替えの後の  $\Delta_j$  の標準偏差および平均をそれぞれ示す。

【0 1 3 0】 図 6 および 7 から分かるように、コードブックを並べ替えてしまうと、危険なコードベクトルは、すべてそのコードブックの最初に配置される。一般に実時間で実施することにより、最初の M 個のコードベクトルに対する正確なエネルギー計算の実行が可能となると仮定すると、エネルギー計算の手順は次のとおりである。

1. 数 3 0 を用いて、 $j = 0, 1, 2, \dots, M$  に対する  $E_j$  の正確な値を計算する。
2. 前記のランコソおよびアタルの自己相関法を用いて、エネルギー

【数 3 5】

推定値

【数 3 6】

8 を使用する。図 4 から、 $M > 1 6$  の場合、エネルギー近似誤差の標準偏差は 1 dB 以内であることが分かる。

【0 1 3 2】 計算上の複雑さという点において、最初の 1 6 個の（危険な）コードベクトルの正確なエネルギー計算には、実証的に約 0. 6 M I P S を要するが、その他の 1 1 2 のコードベクトルに対する偏らない自己相関法では、実証的に約 0. 5 7 M I P S を要する。このように、コードブックのエネルギー計算の全体的な複雑さは、最初の 4. 8 M I P S から 1. 1 7 M I P S まで減少した --- 1 / 4 の縮小である。

【0133】前記のエネルギー計算方法の1つの利点は、DSPのソフトウェア開発の完了後にDSPプロセッサの実時間がどれだけ残っているかによってMを10と128との間のどこにでも選ぶことができると言う点において、容易に倍率調整ができることである。例えば、M=16という初期値を選択しても、実時間で実施して未使用のプロセッサ時間が生じた場合、実時間が不足することなく正確に計算されたコードブック・エネルギー項をより多く得るために、Mを32に大きくすることも可能である。

#### 【0134】後置フィルタ

従来のほとんどのCELP符号器のように、本発明の説明用の実施例による8 kbpsLD-CELP符号器は、図4に示すように音声品質を高めるために後置フィルタを有利に使用している。この後置フィルタは、都合良く長期後置フィルタ、これに続く短期後置フィルタおよび出力利得制御段からなる。短期後置フィルタおよび出力利得制御段は、既に引用したチェンおよびガーショウの論文において提案されたものと本質的に同様であるが、遊休チャネルの効率を改善するために利得制御段が非線形倍率調整の付加的な特徴を備えている点が異なる。一方、長期後置フィルタは、既に引用したチェンの学位論文に説明されているタイプのものである。

【0135】注意を要するのは、符号器においてピッチ周期およびピッチ・タップの閉ループ連帯最適化によって、量子化されたピッチ周期が決定された場合、復号されたピッチ周期が真のピッチ周期と異なる場合があるこ

とである。これは、閉ループ連帯最適化のために量子化されたピッチ周期が開ループ抽出ピッチ周期から1乃至2標本だけ逸れる可能性があるためであり、そのような逸れたピッチ周期は、タップ・コードブックからのピッチ予測器タップのある集合と組み合わせされると、全体的に最も低い知覚的に加重された歪を与えるという理由だけで、非常にしばしば選択される。しかし、これは、復号器の後置フィルタに対しては問題となる。これは、長期後置フィルタが効率的に作用するために真のピッチ周期の滑らかな輪郭を必要とするからである。この問題は、復号器において真のピッチ周期を求める探索を付加的に行うことによって解決される。所望の真のピッチ周期の滑らかな輪郭を回復するには、この単純な方法で十分である。

【0136】第4表から分かるように、後置フィルタは、実施にあたり非常に小さな量しか計算を要しないが、出力音声の知覚的品質には目立った改善を与える。

#### 【0137】実時間での実施

以下の第2、3、および4表により、単一の80 nsのAT&T DSP32Cプロセッサを用いた本発明の諸相によって構築された典型的な8 kbpsLD-CELP符号器の実施の一定の構成面および計算面を説明する。この符号器は、32標本分(4 ms)のフレーム・サイズで実施した。

【0138】次の第2表は、この実施のプロセッサ時間およびメモリ用途を示す。

【表2】

実施モード	プロセッサ 時間  (% DSP32C)	プログラム ROM (kバイト)	データ ROM (kバイト)	データ RAM (kバイト)	メモリ 全体 (kバイト)
符号器のみ	80.1%	8.44	20.09	6.77	35.29
復号器のみ	12.4%	3.34	11.03	3.49	17.86
符号器+復号器	92.5%	10.50	20.28	10.12	40.91

第2表 8 kbps低遅延CELPのDSP32Cプロセッサ時間およびメモリ用途

【0139】この説明のための実施に際し、符号器は、DSP32Cのプロセッサ時間の80.1%をとるのに対し、復号器は、12.4%とるだけである。全二重符号器は、40.91kバイト（または約10kワード）のメモリを必要とする。この数には、DSP32Cチップ上の1.5kワードのRAMも含まれる。この数は、別個の半二重の符号器および復号器に必要なメモリの合計よりかなり低い。これは、符号器および復号器が同一のDSP32Cチップ上で実施されるとメモリを幾らか共有することができるからである。

【0140】第3表は、説明のための8kbpsLD-C

ELP符号器の異なる部分の計算的複雑さを示す。第4図は、復号器に対する同様の表である。符号器のある部分（例えば、ピッチ予測器の量子化）の複雑さは、フレームによって変化する。第3および4表に示した複雑さは、最悪の場合の数（即ち、可能な最大数）に相当する。符号器において、ピッチの周期およびタップの閉ループ連帯量子化は、DSP32Cのプロセッサ時間の22.5%を要し、計算が最も集中する動作であるが、良好な音声品質を達成するために重要な動作でもある。

【表3】

仕 事			No. of DSP32C 4msあたり の 命 令 数	時 間 (80 ns)	MIPS	% DSP32C	
LPC 分析	合成フィルタ	Autocor.	1537	1	0.38	3.07	
		Durbin	481	1	0.12	0.96	
	荷重フィルタ	Autocor.	1581	1	0.39	3.16	
		Durbin	481	1	0.12	0.96	
	対数利得 予測器	Autocor.	141	1	0.035	0.28	
		Durbin	481	1	0.12	0.96	
	励起VQ	コードブック・エネルギー		4672	1	1.17	9.34
		コードブック探索		2970	2	1.49	11.88
	ピッチ 予測器 量子化	期間とタップの連帯最適化		11245	1	2.81	22.49
		ピッチ抽出		4011	1	1.00	8.02
音声検出		562	1	0.14	1.12		
その他		878	1	0.22	1.76		
フィルタ処理およびその他			8063	1	2.02	16.13	

第3表 8kbps低遅延CELP符号器における異なるタスクの計算的複雑さ

【表4】

仕 事			No. of DSP32C 4msあたりの 命 令 数	時 間 (80 ns)	MIPS	% DSP32C
LPC 分析	合成フィルタ	Autocor.	1537	1	0.38	3.07
		Durbin	481	1	0.12	0.96
	対数利得 予測器	Autocor.	141	1	0.035	0.28
		Durbin	481	1	0.12	0.96
後置フィルタ			1832	1	0.46	3.66
フィルタ処理およびその他			1710	1	0.43	3.42

第4表 8 kbps低遅延CELP復号器における異なるタスクの計算的複雑さ

## 【0141】性 能

8 kbpsLD-CELP符号器を同じかそれ以上のビット・レートで動作している他の標準的な符号器と対比して評価し、この8 kbpsLD-CELP符号器が、僅かに1/5の遅延量で同じ音声品質を与えることが分かった。本発明の実施による8 kbpsLD-CELPの4msフレーム版に対し、8 kbpsの伝送チャネルを仮定し、さらにピッチ・パラメータに対応するビットが各フレームで利用できるようになると直ちに伝送されるものと仮定すると、10msに満たない一方向符号化遅延を容易に達成することができる。同様に、8 kbpsLD-CELPの2.5msフレーム版では、6msと7msの間の一方向符号化遅延を、音声品質の劣化も本質的になく、得ることができる。

【0142】低遅延CELP符号器／復号器の実施に関する前記の説明は、大部分8 kbpsの実施という点から進めてきたが、符号器パラメータを幾つか変更することによって、8 kbps以下のビット・レートについても、

本発明によるLD-CELPの実施を行うことができる。例えば、本発明の原理による6.4 kbpsのLD-CELP符号器の音声品質が、最小限の最適化をやり直すだけで8 kbpsLD-CELPのそれと殆ど同様に実現され、すべて以上の教訓から照らして当分屋の実施者の技術の範囲内である。さらに、4.8 kbpsのビット・レートにおいて、フレーム・サイズが4.5ms内外の本発明によるLD-CELP符号器は、30msに及ぶフレーム・サイズの他のほとんどの4.8 kbpsCELP符号器に少なくとも匹敵する音声品質を生成する。

【0143】「化学式等を記載した書面」において例えば「x」に「 $\sim$ 」を冠した表記などは、本文においては「x $\sim$ 」のように記した。

## 【0144】

【発明の効果】以上述べたように、本発明によれば、低ビット・レート低遅延の符号化および復号が可能となる。従来のCELPの僅か1/5程度の遅延で、従来のCELPと同等の音声品質が与えられる。さらに、従来の

技術の複雑さの多くを回避することにより、全二重の符号器が単一のデジタル信号処理（DSP）チップ上に好ましい形で実施できる。さらには、本発明の符号化および復号の方式を用いることにより、ビット誤り率が高い条件の下でも双方向の音声通信を容易に達成することができる。

【図面の簡単な説明】

【図1】従来の技術のCELP符号器を示す図である。

【図2】従来の技術のCELP復号器を示す図である。

【図3】本発明による低ビット・レート低遅延CELP符号器の典型的な実施例である。

【図4】本発明による低ビット・レート低遅延CELP復号器の典型的な実施例である。

【図5】量子化器を備えたピッチ予測器の典型的な実施例を示す図である。

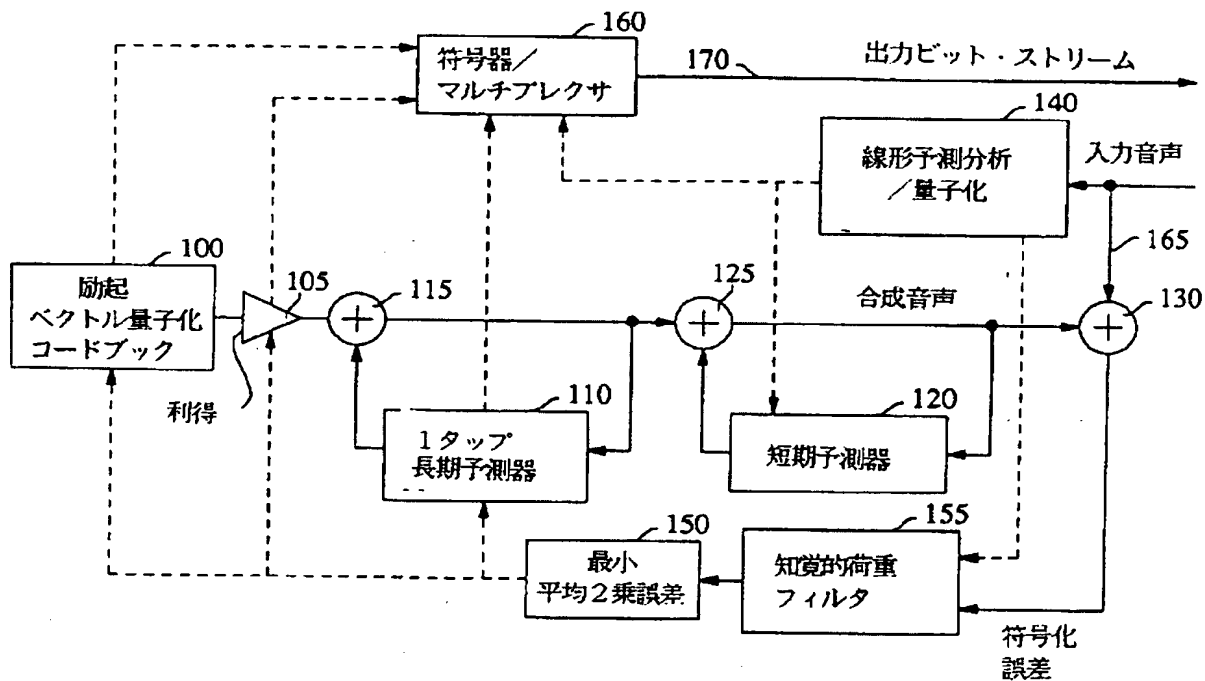
【図6】典型的なコードブックに対するエネルギー近似誤差の標準偏差を示す図である。

【図7】典型的なコードブックに対するエネルギー近似誤差の平均値を示す図である。

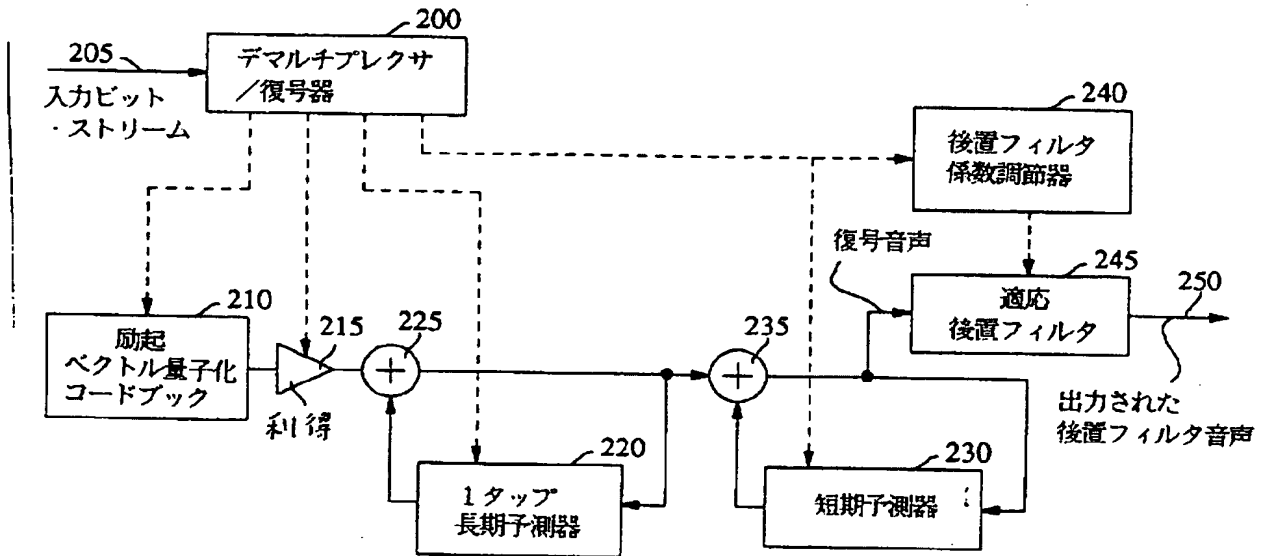
【符号の説明】

- 100、210 励起VQ（ベクトル量子化）コードブック
- 105、215 利得調整要素
- 110、220 1タップ長期予測器
- 115、125、225、235 総和器
- 120、230 短期予測器
- 130 比較器
- 140 線形予測分析／量子化
- 150 最小MSE（平均2乗誤差）要素
- 155 知覚的荷重フィルタ
- 160 符号器／マルチプレкса
- 200 デマルチプレкса／復号器
- 240 後置フィルタの係数調節器
- 245 適応後置フィルタ

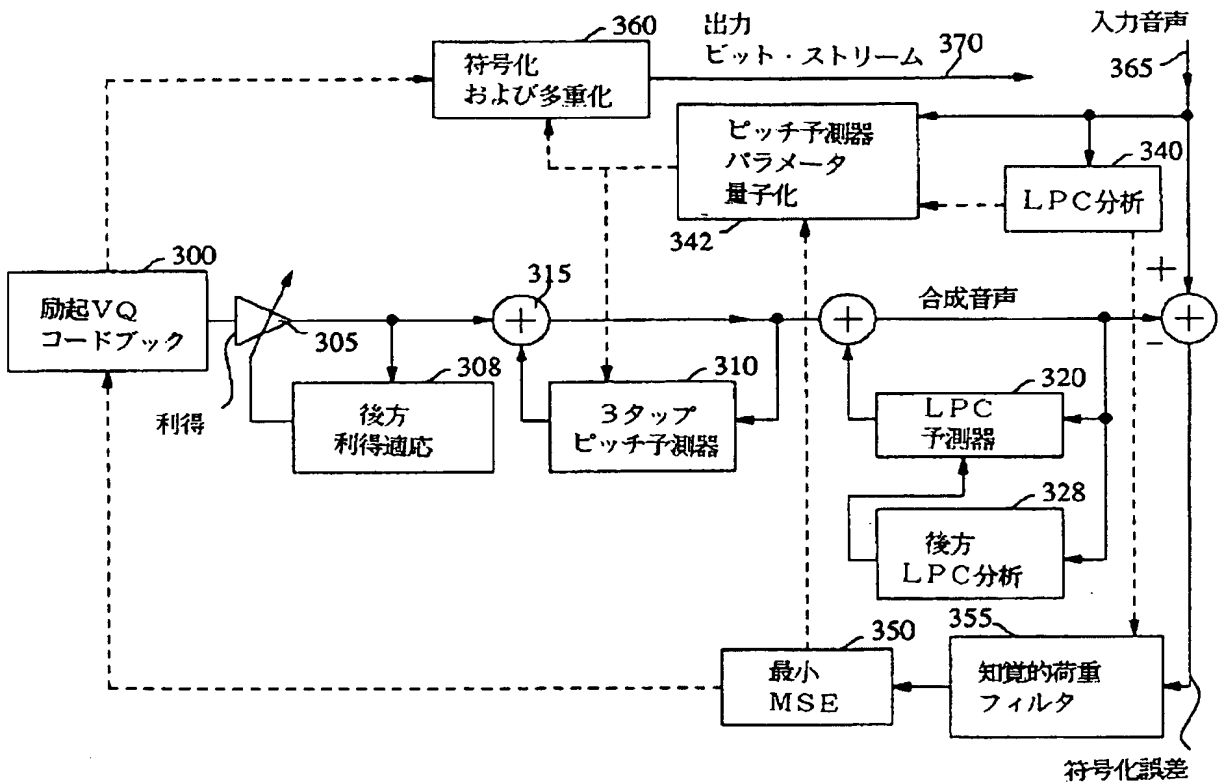
【図1】



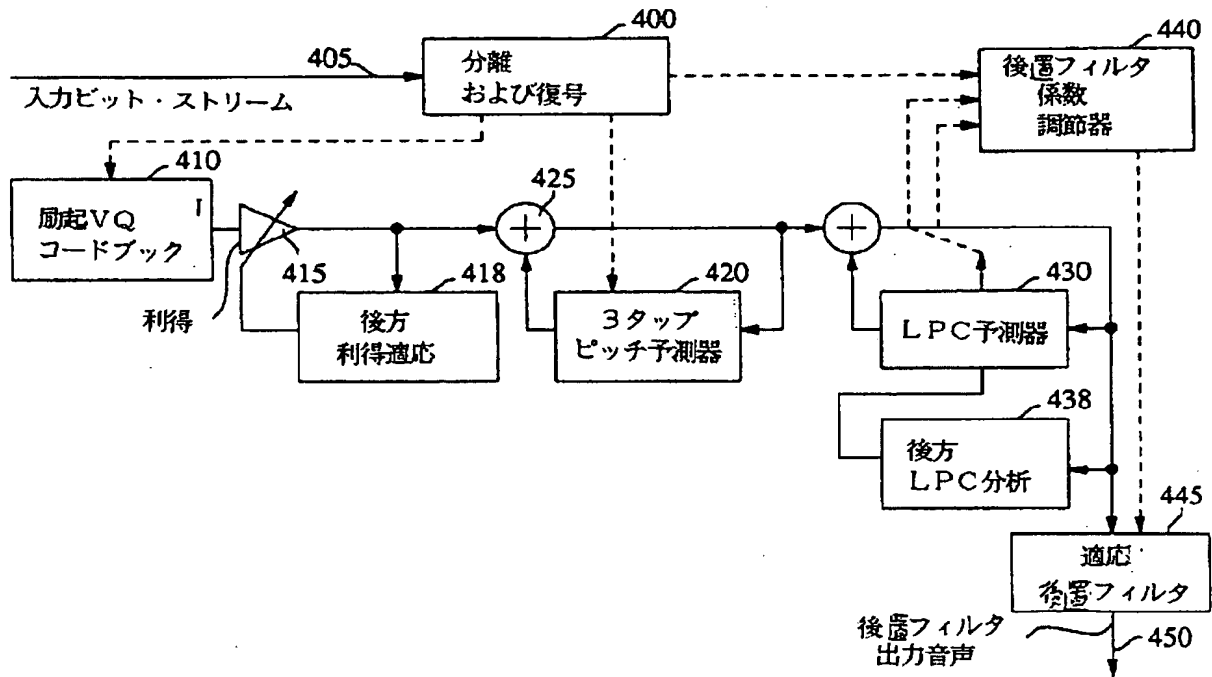
【図2】



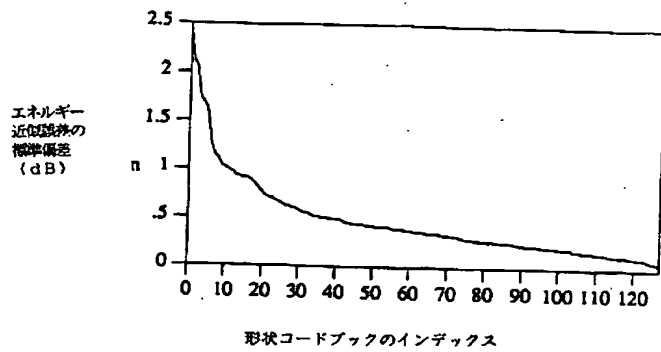
【図3】



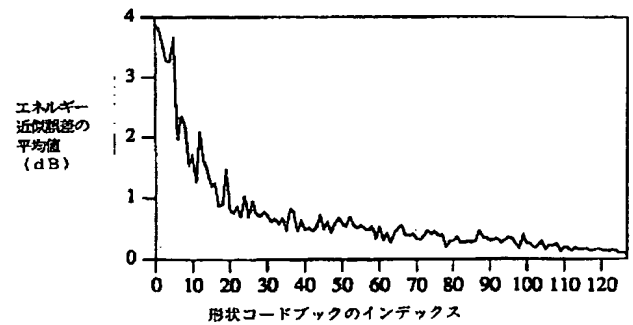
【図4】



【図6】



【図7】



【図 5】

